Mashing up the physical and augmented reality: The Web of Augmented Things Idea

Angelo Croatti
Alma Mater Studiorum – University of Bologna
Computer Science and Engineering Department (DISI)
Cesena, Italy
a.croatti@unibo.it

Alessandro Ricci
Alma Mater Studiorum – University of Bologna
Computer Science and Engineering Department (DISI)
Cesena, Italy
a.ricci@unibo.it

ABSTRACT

Along with Internet of Things (IoT), Augmented Reality (AR) and Mixed Reality (MR) are going to be an important ingredient for the design and development of future smart environments, in particular in enterprise contexts. The vision of Web of Augmented Things (WoAT) – discussed in this paper – aims at exploiting the Web of Things to enable the development of open interoperable pervasive hybrid systems integrating IoT and AR/MR, mashing up both physical Things and *augmented*, AR-based ones.

KEYWORDS

Augmented Reality, Pervasive Computing, the Web of Things, the Web of Augmented Things

ACM Reference Format:
Angelo Croatti and Alessandro Ricci. 2017. Mashing up the physical and augmented reality: The Web of Augmented Things Idea. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Augmented Reality (AR) can be defined as a medium in which digital information is added (superimposed) to the physical world *in registration* with the world itself and displayed to a user dependently on its location and its perspective [1, 7]. Such a registration may occur in different ways, e.g. by means of fiducial markers placed in the environment, perceived and processed through the camera(s) mounted on smart-glasses or devices (or even on smartphones), or directly exploiting the spatial information obtained by sensors mounted on the AR visor. Mixed reality (MR) refers more generally to the merging of real and virtual worlds to produce new environments and visualisations where physical and digital objects co-exist and interact in real time [12]. As defined by P. Milgram and F. Kishino, it is "anywhere between the extrema of the virtuality continuum" [7], that extends from the completely real through to the completely virtual environment with augmented reality (AR) and augmented virtuality ranging between.

AR and MR technologies witnessed a remarkable boost in recent years. Commercial technologies – examples are Wikitude [1], Apple ARkit[2], Google ARCore[3] — allow for creating indoor and outdoor applications where physical places and objects are decorated by dynamic virtual information and objects that can be perceived by users simply using a tablet or smart-glasses. Mixed Reality technologies such as Microsoft Hololens[4] and Meta2[5], allow for mixing physical environments with fairly sophisticated *holograms* that users can perceive and interact with by means of wearable AR visors.

AR and MR technologies can have a huge impact from an application point view, allowing for reshaping the environments where people work and live, rethinking the way in which they interact and collaborate. In that perspective, it is interesting to consider how these technologies can be put in synergy with pervasive computing/ubicomp technologies [11, 15], and then Internet of Things (IoT) and Web of Things (WoT) as main ingredients of modern smart environments and spaces [2, 6, 9].

At a first glance, AR could be exploited to design more immersive User Interfaces (UIs), with UI elements escaping out of 2D glass screens to be flexibly placed and perceived in the real-world, eventually shared by and interacting with multiple users, inhabiting the same physical space. The "Milgram–Weiser" chart depicted in Figure 1 (taken from [12]) arranges ubiquitous computing and AR technologies in that perspective.

Beyond that view, AR and MR could be devised as technologies bringing a form of deeper *augmentation* [5, 10], allowing the design and development of smart environments where the physical space is augmented with functionalities based on both computational devices spread in the physical environment and virtual entities, *holograms* spatially located in the same environment as well. In that view, a smart environment is designed in terms of an open and dynamic collection of both physical and virtual/*augmented things* (in the IoT perspective), eventually interacting together and more generally with other software agents and systems through the network.

This calls for devising – first of all – a proper conceptual framework that makes it possible to understand the integration of pervasive/ubiquitous computing and AR/MR at a proper level of abstraction. Then, at a more technological level, a main issue which is going to affect also AR and MR open/large applications is *interoperability*, analogous to what happened for IoT.

[1]https://www.wikitude.com
[2]https://developer.apple.com/arkit/
[3]https://developers.google.com/ar/
[4]https://www.microsoft.com/microsoft-hololens/en-us
[5]https://www.metavision.com/

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

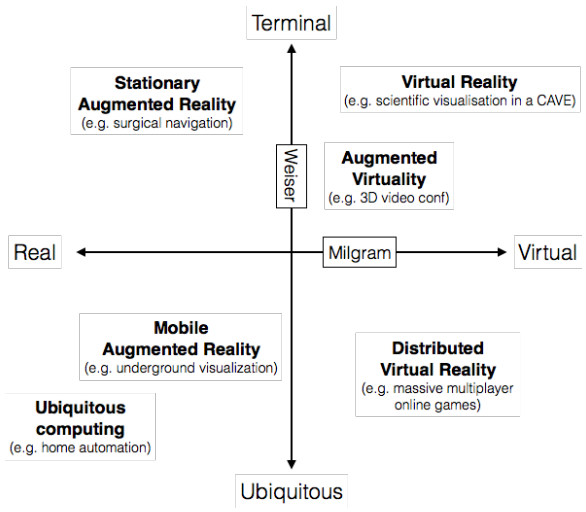

Figure 1: The Milgram-Weiser chart about the relationships of user interface paradigms (taken from [12], pag. 31)

To tackle these issues, the Web of Augmented Things (WoAT) idea discussed in this paper aims at defining an interoperable application layer enabling the mashing up of physical and AR/MR entities exploiting the Web of Things [4]. The WoAT idea relies on a conceptual framework called Augmented Worlds [3, 10], aiming at capturing the integration and synergy between AR and pervasive computing in terms of a *bi-directional* augmentation.

The remainder of this paper is organised as follows: in Section 2 we provide a very brief background on Augmented Worlds, focusing on the motivations that bring to WoAT and the contribution discussed in this paper. Then in Section 3 we discuss the WoAT idea, focusing on its relationship with WoT and the benefits bringing into the WoT panorama. In Section 4 we briefly discuss the design and implementation of a first infrastructure devised for playing with the WoAT idea. Finally, we conclude the paper by briefly discussing in Section 5 the challenges that the WoAT brings and then sketching a roadmap about future work.

2 THE AUGMENTED WORLDS CONCEPTUAL FRAMEWORK

The conceptual framework of Augmented Worlds [3, 10] aims at introducing a conceptual model capturing main aspects of augmented technologies and their integration/convergence. In general terms, an *augmented world* (AW) can be defined as a software application that enriches the functionalities of a particular physical environment (e.g., a room, a building, a city, etc.) by means of full-fledged computational objects – encapsulating a state and a behaviour – referred as *augmented entities*.

An augmented entity is situated in a specific location of the physical space and can be equipped with an AR representation, an *hologram*, that can be perceived by humans by means of proper AR devices. Besides, any augmented entity is meant to have a proper API that makes it possible for any software agent[6] to interact with it, by observing its observable state and by executing actions on it.

An augmented entity can be either totally virtual or *coupled* to some concrete physical object, representing a kind of virtual extension *or mirror* of the object in the physical world [10]—including also *human users*, enabling agents to observe users' actions and movements. This view promotes a kind of *bi-directional* augmentation: on the one hand, augmented entities can be framed as an augmentation of the physical environment; on the other hand, the physical environment can be framed as an augmentation of the augmented entities, so that the state/behaviour of augmented entities can represent properties and phenomena occurring in the physical world.

The AW view promotes the design and development of smart environments as multi-user open systems composed by dynamic and heterogeneous set of augmented entities, interacting with users possibly adopting different kind of AR technologies. In that perspective, *interoperability* is going to be a major issue even in these hybrid physical augmented systems, like in the case of pure IoT systems. The WoAT idea is about exploiting the WoT model and principles to tackle this issue, by defining an interoperable application layer that – on the one hand – allows for the engineering of open/complex AWs using different platforms and different AR technologies, and – on the other hand – makes it possible to extend WoT systems with AR capabilities.

3 FROM WOT TO WOAT

The basic idea is to model every augmented entity as a *thing* in the WoT perspective, featuring:

- a root URL, that corresponds to its network address.
- a REST-ful Web API, based on self-descriptive messages, HTTP operations (GET, POST, PUT, DELETE, HEAD) and event-oriented mechanisms (e.g., web sockets).

In the following, we will refer to these augmented entities as *augmented things* (AuTs) — as things in the WoT perspective that represent augmented entities living in an AW. Some AuTs could be totally virtual, while being situated in some location of the physical space, others could be coupled with some physical objects. Such objects could be either simple physical objects, or things of a WoT: in that perspective, WoAT is about WoT systems augmented by an AR layer.

The model of an AuT can be defined as a specialisation of the Web Thing model as defined in one of the recent proposals [14] so as to include AW concepts and the augmented entity model. Accordingly, the root resource model of an AuT can be framed as:

{aut}	AuT root resource URL
{aut}/model/	The model of the AuT
{aut}/properties/	The list of properties
{aut}/properties/{id}	A specific observable property
{aut}/actions/	The list of actions
{aut}/actions/{id}	A specific action
{aut}/actions/{id}/{actionId}	A specific execution of an action

[6] For "software agents" here we mean any kind of autonomous software component, encapsulating its logical control flow

Observable properties – which define the observable state of an augmented entity [3], following the A&A conceptual model [8] – are naturally mapped onto properties as defined by the WoT. Analogously, actions (operations) – which define the mean that agents have to act upon and have an effect on the entities – are naturally mapped onto actions of the WoT model. Then, aAs is in the case of the Web Thing model:

- GET can be used to retrieve (observable) properties of an Augmented Thing (i.e., an augmented entity).
- POST can be used to request the execution of an action over an AuT.

Besides, the AuT model includes specific features/properties that come from the AW conceptual model:

{aut}/properties/location	location in the physical world
{aut}/properties/orientation	orientation in the physical world
{aut}/properties/extension	extension in the physical world
{aut}/hologram	representation in AR

The location property contains the current location in the physical world/space of the AuT, defined in the system of reference defined by the region characterizing the AW. The location is defined also in the case of marker-based AR-systems, where marker are exploited to define the system of reference where the augmented entity are immersed. Each AuT has a dynamic orientation in the 3D space, which is represented by the orientation property. The extension property defines instead the extension (geometry) of the AuT as the set of points occupied in the space—that can be possibly composed by a single point. Finally hologram in the AuT root includes the information required to create the AR representation on the human user side.

To support event-oriented interactions, AuTs can be subscribed, both at the resource level and the property/action level, like in the Web Thing model proposed in [14]. Subscription allows for agents observing the AuT to be notified via a push event whenever the state of an observed resource changes.

Besides the augmented things, an augmented world itself is modelled as a WoT Thing, featuring the following model:

{aw}	Augmented World URL root
{aw}/properties/region	physical region
{aw}/properties/things	list of augmented things
{aw}/properties/places	list of places to be tracked
{aw}/properties/places/id	specific place
{aw}/properties/places/id/things	things located in that place

The region property describes the region of the physical world covered by the AW, including the system of reference used for locating the AuT. things describes the dynamic set of augmented things currently part of the AW. Dynamic lookup and discovery of AuTs occurs by GETs (or subscriptions) on this property, possibly specifying filters. Instead, dynamic creation and removal of AuTs occur by means of POSTs on this property.

The places property is introduced to support a main functionality on the agents' side, which is the continuous observations of physical subregions of an augmented world so as to be notified about events related to augmented entities entering/exiting the subregion. An agent can create a *place* (using a POST on places) specifying a unique identifier (e.g., kitchen) and a physical region (which must be a subregion contained in the AW). Given an a new place, a property things keeps track of the things that are currently located in the corresponding subregion. By subscribing to the things property, an agent can perceive when an augmented thing entered the place or left it.

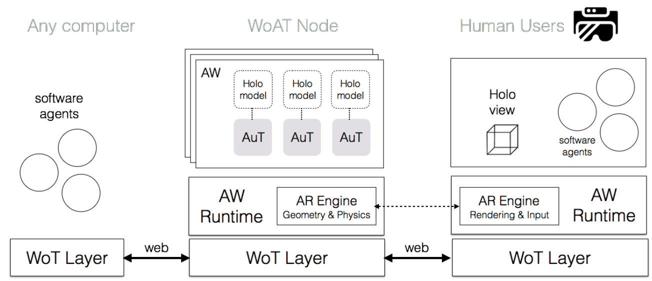

Figure 2: WoAT Architecture – An Abstract Coarse-Grained View

To summarise, the key points of the WoAT idea are:

- Enabling the development of open and interoperable systems integrating IoT and AR/MR.
- Injecting a deeper view on *augmentation* about AR (for IoT), beyond UI, following the Augmented Worlds conceptual model.

4 A FIRST PROTOTYPE PLATFORM TO PLAY WITH WOAT

To start playing with the WoAT idea, we designed and implemented a first prototype platform/infrastructure, for developing and executing Augmented Worlds based on WoT. Figure 2 shows an abstract view of the architecture. A WoAT node is any server side node meant to host the execution of one or multiple instances of augmented worlds, running on top of the Augmented World Runtime (AW Runtime). The WoT Layer provided the Web API based on the WoAT model – compatible with WoT – described in previous sections, dispatching the requests to the AW Runtime. AW Runtime manages the state and behaviour of augmented entities, as well as their dynamic creation/disposal, and the hologram model associated to each AuT. Every time the state of an AuT is changed, the hologram model is updated, according to the strategy implemented by the specific AuT. The AR Engine inside the AW Runtime is the component responsible of managing the holograms at the augmented world level (geometry and physics, not rendering).

Figure 2 shows also (on the right side) the software stack running on wearable/mobile devices on the user side (e.g., AR-enabled smart-glasses). In this case, the AW Runtime is not managing anything about augmented entities but the AR-based view/rendering of holograms, that depends on the current position and orientation of the human user inside the physical space where the AW is located. The AR Engine on this side is responsible of rendering and tracking user position and input. The AW Runtime includes also API that allows software agents to interact with an AW. This API is actually

```
@HOLOGRAM("LampHologram")
class Lamp extends AugmentedThing {

 @PROPERTY(onUpdate = "onStatusUpdate")
 private String status = "off";

 @ACTION
 public void turnOn() {
  customProperty("status", "on");
  link("realLampId").sendMessage("Turn on!");
 }

 @ACTION
 public void turnOff() {
  customProperty("status", "off");
  link("realLampId").sendMessage("Turn off!");
 }

 void onStatusUpdate() { log("New lamp status: " + customProperty("status")); }
 public void onConnected(String id) { /* ... */ }
 public void onDisconnected(String id) { /* ... */ }
 public void onMessageReceived(String id, JsonObject msg) { ... }
}
```

Figure 3: An example of an AuT implementation.

based on the WoT layer, which enables the interaction with the WoAT node using WoT protocols and conventions.

Finally, any (authorised) software agent – running on any computing devices on the Internet – can join and work inside an AW, through the WoT layer (Figure 2, left).

An ongoing prototype implementation is available online [7]. The AW Runtime in the WoAT Node is written in Java, using the Vert.x framework for implementing the WoT layer. The AR Engine is based on Unity 3D equipped with the Vuforia plugin, with AW scripts developed in C#. The AW Runtime and the AR Engine interact by means of sockets as IPC.

Currently a Java-based API is provided for developing the templates of the augmented entities that can be created and executed inside an AW. Figure 3 shows a simple example, based on a virtual light, which can be switched on/off and linked with a real lamp. The state and behaviour of an AuT is represented by a class extending the `AugmentedThing` base class. Methods annotated with `@ACTION` are bound to actions that can be triggered through the Web API, while properties are defined by `@PROPERTY` annotation. The AW Runtime adopts an event-loop to accept and dispatch the requests about action execution, causing the execution of a proper method. Instances of these templates represent the concrete AuT actually available inside an AW. Besides this Java part, the hologram model associated to an AuT can be defined by means of a class annotation `@HOLOGRAM`, specifying the identifier of a Unity 3D prefab. The template for an Augmented World is defined by a JSON configuration file, including information about the physical region where the AW is mapped, the initial set of AuT to be created (along with their parameters) and other details useful for setting up the world.

5 ROADMAP FOR FUTURE WORK

The vision of WoAT brings a large of set of open issues and challenges. Besides those that concern IoT and AR per se – e.g., about reliability, privacy and security, performance – here we focus on two main ones that specifically concern the engineering of AWs on top of WoT.

The first is about scalability. WoT underlying distribution model – based on the Web and REST – fully supports scalability. Conversely, the simplifying assumption made for the architecture of the prototype infrastructure presented in Section 4 – i.e., a whole augmented world runs on a single WoAT node – is clearly problematic as soon as we consider large-scale AWs. These make it not feasible to host the execution of all the augmented entities in a single node. So a main challenge is about the design of an architecture capable of scaling with the complexity of the augmented worlds, eventually exploiting multiple WoAT nodes to host a single AW or adopting a cloud-oriented design, i.e. using cloud-based PAAS services for storing large sets of augmented things and for executing actions on them as tasks.

The second main challenge is about the development of effective *multi-user* applications and environments, which is a common case for AWs, that is: augmented worlds where multiple human users share and interact with the same physical and AuTs. This introduces real-time constraints that are required to guarantee a certain level of synchronisation and coherence in what is perceived by users and the state of the environment, including the state of both physical Things and Augmented Things, possibly running on different nodes (see previous point). Building effective multi-user AW capable of scaling with respect to the number of users concurrently working inside the AW is challenging. Existing research works on collaborative multi-user online systems (e.g., Croquet [13]) will be an important reference to tackle this issue.

REFERENCES

[1] Ronald T. Azuma. 1997. A Survey of Augmented Reality. *Presence: Teleoper. Virtual Environ.* 6, 4 (Aug. 1997), 355–385.
[2] Mark Billinghurst, Adrian Clark, and Gun Lee. 2015. A Survey of Augmented Reality. *Found. and Trends in Human-Computer Interaction* 8, 2-3 (2015), 73–272.
[3] A. Croatti and A. Ricci. 2017. Towards the Engineering of Agent-Based Augmented Worlds. In *5th Int. Workshop on Engineering Multi-Agent Systems (EMAS'17), held at AAMAS 2017.*
[4] A. Croatti and A. Ricci. 2017. Towards the Web of Augmented Things. In *2017 IEEE International Conference on Software Architecture Workshops (ICSAW).* 80–87.
[5] Douglas C. Engelbart and William K. English. 1968. A Research Center for Augmenting Human Intellect. In *Proc. of the Fall Joint Computer Conference (AFIPS '68 (Fall, part I)).* ACM, New York, NY, USA, 395–410.
[6] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. 2014. Industry 4.0. *Business & Information Systems Engineering* 6, 4 (01 Aug 2014), 239–242.
[7] P. Milgram and F. Kishino. 1994. A Taxonomy of Mixed Reality Visual Displays. *IEICE Trans. Information Systems* E77-D, 12 (Dec. 1994), 1321–1329.
[8] A. Omicini, A. Ricci, and M. Viroli. 2008. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 17 (3) (2008).
[9] Kasey Panetta. 2017. Exploring Augmented Reality for Business and Consumers. (2017). Gartner report.
[10] A. Ricci, M. Piunti, L. Tummolini, and C. Castelfranchi. 2015. The Mirror World: Preparing for Mixed-Reality Living. *IEEE Perv. Comp.* 14, 2 (Apr 2015), 60–63.
[11] M. Satyanarayanan. 2001. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications* 8 (2001), 10–17.
[12] D. Schmalstieg and T. Höllerer. 2015. *Augmented Reality: Principles and Practice.* Addison-Wesley.
[13] D. A. Smith, A. Kay, A. Raab, and D. P. Reed. 2003. Croquet - a collaboration system architecture. In *First Conference on Creating, Connecting and Collaborating Through Computing, 2003. C5 2003. Proceedings.* 2–9.
[14] Vlad Trifa, Dominique Guinard, and David Carrera. 2015. Web Thing Model - W3C Member Submission. (2015). Available at: https://www.w3.org/Submission/wot-model/.
[15] Mark Weiser. 1999. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3 (July 1999), 3–11. Issue 3.

[7] https://bitbucket.org/account/user/awuniboteam/projects/AW