A step towards more eco-responsible computing

Richard Fontaine, Rémy Courdier, Denis Payet

A step towards more eco-responsible computing

Fontaine Richard[1][0000−0001−9457−3261], Courdier Rémy[1], and Payet Denis[1]

Laboratoire d'Informatique et de Mathématiques (LIM), University of Reunion Island, France

Abstract. On a global scale, the environmental footprint of digital technology represents a continent two to three times the size of France and five times the size of the French car fleet. In order to limit the negative impact of this overabundance and to optimize the emerging computing potential of our environment, we propose an architecture model and its implementation in order to allow the mutualization of the components embedded in our connected devices.

Keywords: Agent · Ubiquitous computing · Eco-computing

1 Introduction

We are at the beginning of a new computing era which is manifested by a trend to integrate computing into the physical objects of everyday life [24, 2, 21]. While providing a lot of benefits, this technological advance brings with it a set of environmental issues [3, 4].

According to the European Union [6], in 2020, there were 30 critical raw materials (against 14 in 2011) and among them, many elements are directly linked to new technologies: Cobalt (lithium-ion batteries); Germanium (optical fibers); Hafnium (processor); Indium (touch screen); Tantalum (liquid crystal displays, dynamic random access memory (DRAM) chips), or Lithium (battery). These elements were generally related to the manufacture of computers, printers and other common digital objects. But since 2015, we notice that a shift has taken place [3] :

- Televisions represented 5 to 15 % of the impacts in 2010 against 9 % to 26 % by 2025;
- Smartphones represented 2 to 6 % of the impacts in 2010 against 4 to 16 % by 2025;
- Connected objects represented 1 % of the impacts in 2010 against 18 % to 23 % by 2025.

Based on these statistics, we can see that the depletion of our global reserves is fast approaching. We therefore need a new vision of eco-design that focuses not only on the energy impact, but on a direct safeguard of our raw materials.

To do this, we believe it is necessary to see in our connected objects more than their initial functionalities. These hidden and currently unexploited functionalities, which we will call dispositions, represent the core of our approach. We

propose, through an architecture model, to reveal them and to mutualize them virtually in order to allow the emergence of new services without systematically adding new physical connected objects. This model has been implemented within an Android framework named Agent Framework For Omnipresent Real Device (AFFORD).

Before highlighting the technical solutions that are used to make this mutualization of dispositions, we will illustrate our concept in order to demonstrate its advantages.

1.1 Example of a mutualisation of components: a monitoring agent

In this example; we will consider a monitoring agent A which has the objective Ω of alerting in case of a fall. This objective can be divided into three required objectives ω which are:

- ω_1 : to detect a fall
- ω_2: to alert the user by a sound
- ω_3 : alert the user with a flashing light

In our case, the agent has access to a set of connected devices including: a light bulb, a computer, and an accelerometer. So, we can describe the Obj objects of the environment through the n dispositions they present:

- Obj^1 : The accelerometer
  - n_1 detect a fall
- Obj^2 : The computer
  - n_2 produce sound
  - n_3 display a picture
- Obj^3 : The connected light bulb
  - n_4 make light

In a usual context, the service would be managed by a single entity, whereas in our case, it is possible to dissociate the monitoring service, the capture of information and the response of the system. When the system is started, agent A seeks to accomplish the Ω objective. Because of the presence of the accelerometer Obj^1 in its environment and its disposition n_1 to detect a fall, the objective ω_1 is made permanently active. If a fall is detected, the agent reacts and its ω_2 and ω_3 objectives become active. In an optimal case, where the light bulb Obj^3 and the computer Obj_2 are accessible, the agent has the possibility to activate the behaviours ω_2 and ω_3. In this way, attention is focused on the problem of falling. In the same way, it is interesting to note that this same light bulb Obj^3 and this same computer Obj_2 can be used as an alert source for other types of monitoring (such as intrusion, temperature, or hydrometry monitoring). This vision of our environment thus allows us to reduce redundancies within a system by privileging a more intelligent and reasoned use of the objects that surround us.

2 An ecological challenge, but also a technical challenge

To make this vision real and to start this change of habit, three points must be taken into account:

From a conceptual point of view , we believe that it is necessary to propose a high-level solution, which would allow a quick deployment on a maximum of existing devices.

From a technological point of view , considering the ubiquitous aspect, the solution must be light enough to allow its deployment on devices with low computing capacity.

From a human point of view , its use must be as natural as possible. To achieve this, we will emphasize the presence of autonomous entities that can decide individually on the provisions they wish to share.

2.1 Current technological possibilities

Before presenting our solution, we will examine the existing technical possibilities that can be used to try to solve our problem. In order to do this, we will look at three different levels of abstraction : middleware, agents, and artifact concept.

Middleware. In ubiquitous systems, the middleware is generally considered as a generic layer which provides basic functions [1, 12]. This approach is commonly used [11, 16], however a limit to the use of middleware comes with the fact that, usually, a system has as many middlewares as there are communication problems [14, 18]. Consequently, the more heterogeneous entities are involved in a system, the more it may contain a large number of middleware in order to hide the communication problems.

Even if some authors propose the implementation of a "middleware of middlewares" in order to provide a unique interface to the applications [23], the system will be unable to adapt as new technologies appear.

Agents. In an ambient intelligence context, an option to overcome the heterogeneity of the environment is the use of agents ([9, 13, 22, 25]). However, if we want to scale up, it is inconceivable to put the responsibility of describing all the connected objects and their possibilities on the level of the agents in charge of the services. Based on this observation, some proposals try to overcome the problems of making connected objects available by linking them to specialized "agent-objects" [17, 15, 20]. Unfortunately, this option is confronted with a conceptual problem. Indeed, it is not advisable to model an object using the agent concept, because it does not have its characteristics [5, 7, 13, 25].

Artifacts. The A&A (Agent & Artifact) meta-model is characterized by three main abstractions [19]:

- Agents: proactive components of systems, encapsulating the execution of activities in a given environment;
- Artifacts: passive components of systems, such as resources and media, intended to be used by agents to support their behaviour;
- Workspaces: conceptual containers of agents and artefacts, useful for defining the topologies of the environment and notions of the locality.

The concept of artefact, due to its abstract nature and not being conceived for a specific purpose, unfortunately does not directly solve the problem of the mutualisation of the dispositions of objects present in our environment. However, its coherence in terms of agent/object modelling and its polymorphic nature make it an interesting theoretical tool for our model proposal.

3 Concept description

From the constraints and elements raised above, we can deduce that the current solutions do not meet the requirements of our proposal. From this observation, we propose to modify our vision of the environment by proposing a virtual decomposition of our objects [10] in the form of a dispositional artifact that we will define as follows:

Definition 1 (Decomposition into dispositional artifacts). *Each object Obj, can be represented by a set of artifacts Art according to the usage such as :*

$$Obj -> \{Art_n\}$$

with Obj a connected object, Art a virtual artifact, n the disposition of the object encapsulated by the artifact.

From this decomposition, it is then possible for the agent to perceive new possibilities within its environment that we will call dispositional opportunities.

Definition 2 (Emergence of dispositional opportunities). *Let W (an agent-object system) $= (A, Obj)$ and W_p (an agent-artifact system) $= (A, Art_p)$ composed of an artifact Art with a disposition p. A dispositional opportunity exists if and only if there exists an opportunity τ, such that :*

$$1.\ W = (A, Obj) \text{ does not possess } \tau$$

$$2.\ W_p = (A, Art_p) \text{ possess } \tau$$

and $Art_p \in Obj$

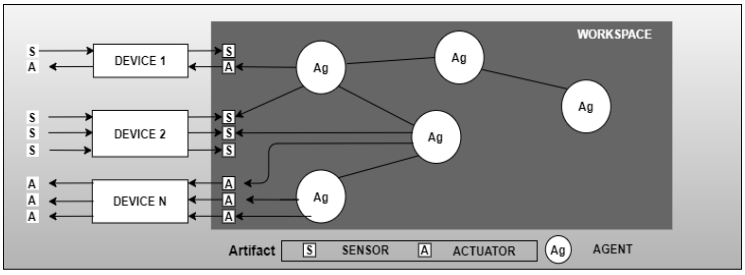

Fig. 1. Mutualisation of components within a workspace

Based on these definitions, we propose an architecture model that involves three types of entities based on the A&A concept: agent, workspace and artifact. Within this model, the layouts are encapsulated in artifacts that will be used as virtual media to make the dispositions of connected objects accessible to an agent (Fig.1).

Because the roles of the agent and the workspace are globally unchanged, we will directly focus on the conceptual additions related to the decomposition of the objects into a dispositional artifact tree.

3.1 Architecture model

We will therefore focus on the seven main artifact classes: Artifact, Composite Artifact, Main Artifact, Final Artifact, Sensor Artifact, Actuator Artifact, Service Artifact.

- Artifact
  This class is the primary component of the layout decomposition. Globally, it defines all the primitives useful for programming the observable behaviours of artefacts in accordance with the frameworks or implementation environments. This first class allows the establishment of the dependency link between artefacts, as well as its observable state within workspaces.
- Composite Artifact
  The CompositeArtifact class is mainly an artifact container. Its role is to keep a trace of the structure of the original connected object. It is also possible that it is itself under the management of an artifact if it is also a component element.
- Main Artifact
  The Main Artifact class virtually represents the object and contains information about it (nature, state, mac address, etc.). It is also responsible for accessing the artifacts and only this class is directly linked to the workspaces.
- Final Artifact
  The elements of this class represent the artifacts that are directly accessible in the workspace. All classes that inherit from it (Sensor Artifact, Actuator

Artifact, Service Artifact) can be seen as the leaves of the decomposition tree of the original object and therefore cannot have artifacts that compose them.

3.2 Properties

Thanks to the artifact classes seen previously, it becomes possible to virtually describe the dispositions of an object in the form of a tree. Within this description, the leaves represent the object's dispositions and the internal nodes, the structure of its decomposition.

The Figure 2 illustrates a virtual decomposition of a smartphone into a tree where the root represents the original object (Main Artifact), the intermediate nodes the structure of the decomposition (Composite Artifact) and the leaves the dispositions that can be made accessible (Final Artifact).

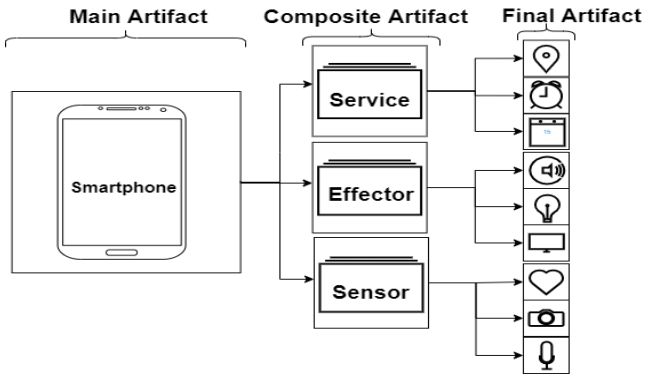

Fig. 2. Example of decomposition

As a result, the object is no more represented in a monolithic way, but is decomposed into a set of dispositional artifacts. It then becomes possible to mutualize them within workspaces in order to propose services that fully use the possibilities of our connected environment. From a conceptual point of view, unlike the use of agents, mutualization is based on a coherent paradigm, and their uniformity and high level of abstraction avoid the overabundance of middleware. The use of artifacts also has the advantage of solving two key problems related to the Publish/Subscribe concept, which is generally used to define the modalities of transmission of sensor information. First, the problem of efficiently linking an event with many subscribers on one type of event can be solved by "relocating" the associated Sensor object to another device. The connected object will only have to manage one Subscriber per sensor. Broadcasting to interested entities will be done through the Sensor artifacts. Second, the problem of efficient multicasting of events within a network of event providers is solved by having the

agent connected directly to the desired sensor. Thus, interested entities will only have access to information that is relevant to them.

4 Android implementation

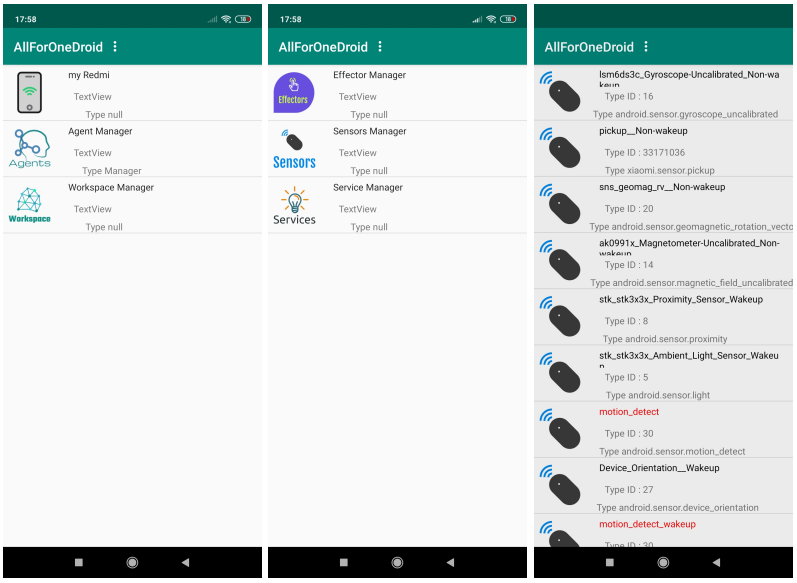

Fig. 3. An application on the Android platform

An implementation of this model, named Agent Framework For Omnipresent Real Device (AFFORD), has been developed on Android through a framework in order to show the feasibility of the model. AFFORD is based on the "Software Kit for Ubiquitous Agent Development" (SKUAD) platform, which is a platform for creating ambient agents, being able to manage sensors and effectors, whose performances and design allow an embedded use [8]. The application we made, in APK format (9.13MB), can detect all sensors embedded in an Android device and currently offers seven effectors and two services (Fig.3).

At runtime, if a component is available and the authorizations allow it, the system proceeds to the creation of artifacts as an instance of the associated class. The user can then choose, via the user interface, which artifacts to make available and which agents to activate within the workspace.

During the test phases, AFFORD was tested on different Android devices (smartphone and tablet) in order to highlight the presence of provisions. This first step confirmed that many possibilities were hidden in our smartphones. In addition to the effectors and services that can be implemented (e.g. ringtone,

voice recognition, buzzer.), we can see in the table (Fig.4) that natively they were able to propose at least twenty artifacts.

Brands	Samsung	Motorola	Samsung	Samsung	Xiaomi
Model	Galaxy s8	One zoom	SM-T870	SM-N960F	Redmi Note 8T
RAM (Go)	4	4	8	6	4
SensorArtifact	26	44	20	27	22

Fig. 4. Some examples of tested smartphones

4.1 Example in real environment

A real-world example of this implementation has been proposed through a falling detector. The devices used in this test are: a Samsung Galaxy S8 smartphone, a Samsung Tab A (2016) tablet and an Acer TravelMate P257 laptop. At the end of the layout analysis step, the smartphone offers twenty-six Sensors and the tablet offers five.

To perform this test, we used the artifact encapsulating the accelerometer, present in any smartphone and usually used to determine the orientation of the screen or to stabilize photographs, as a shock sensor. It should be known that a smartphone accelerometer can generally detect movement in the micrometer range. Moreover, in order to show that the possibilities are not limited to the components of the device, we have created an artifact encapsulating synthesis and voice recognition to perform a simple dialogue with a user.

Thus, the intelligent entity, in this implementation a SKUAD agent, becomes able, by adjusting the sensitivity parameters, to discriminate different types of shocks (fall, heights, direction, presence of a final shock). In case of shock, the agent will try to attract the user's attention via the flash or the vibrator of Android devices or by making the computer screen blink. If the agent does not get a response, he can then use voice synthesis and voice recognition to talk to a user. In this case, the agent asks a series of closed questions giving the choice to the user to answer among a set of predefined answers. Each answer pronounced by the user will be transformed into text and then analyzed by the agent in order to make a report of the situation.

4.2 Encouraging results

Through the previous example, however simple, we find ourselves virtually and schematically in a system with 3 processors, 3 batteries, 12 GB of Ram and 580 GB of storage. This power is much higher than the one required in a current use. This particularity shows us that the sensing and broadcasting entities can be externalized without the necessity of adding components with a processing capacity. We have also shown that the current dynamics seeking to make the

environment intelligent could pass, not by a systematic integration of computing power within each object, but rather by a more reasoned and mutualized use of the possibilities of our devices. Moreover, due to the fact that processing can be done at the peripheral level, this solution offers an alternative to cloud computing. It also allows to reduce the energy footprint, since it does not require the internet network or the datacenter, while limiting the leakage of personal data.

5 Conclusion

The current trend is to systematically integrate computing capabilities into each object. However, an alternative would be to propose a more reasoned and mutualised use of the possibilities of our devices. Thanks to an architecture model inspired by the concept of artefact, we propose to virtually put forward the dispositions of our connected devices in order to reduce the redundancies within an environment. We implemented the model on Android which allowed us to highlight the multitude of unused opportunities present in our environments. From the results, we found that a single smartphone or tablet could power many services. This collective power allows us to reduce the number of batteries, screens and computer components (RAM, processor, storage) and thus contribute to a step towards more eco-responsible computing.

References

1. Baldauf, M., Dustdar, S., Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing **2**(4), 263–277 (2007)
2. Böhlen, M., Frei, H.: Ambient intelligence in the city overview and new perspectives. In: Handbook of ambient intelligence and smart environments, pp. 911–938. Springer, Boston, MA (2010)
3. Bordage, F.: Study - The environmental footprint of the digital world (2019)
4. Cailloce, L.: Numérique: le grand gâchis énergétique (2018)
5. Cetnarowicz, K.: From algorithm to agent. In: Allen, G., Nabrzyski, J., Seidel, E., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) Computational Science – ICCS 2009. pp. 825–834. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
6. Commission, E.: Eu commission, critical raw materials, internal market, industry, entrepreneurship and smes. `https://ec.europa.eu/growth/sectors/raw-materials/specific-interest/critical_en` (May 2020)
7. DeLoach, S.A.: Multiagent systems engineering: A methodology and language for designing agent systems. Tech. rep., Department of Electrical and Computer Engineering of Air Force Institute of Technology (1999)
8. Denis, P.: Skuad, software kit for ubiquitous agent development. http://skuad.onover.top/ (2018), `http://skuad.onover.top/`
9. Ferber, J.: Les systèmes multi-agents: vers une intelligence collective. I.I.A. Informatique intelligence artificielle, InterEditions, France (1995), `https://books.google.com/books?id=jlpDOwAACAAJ`

10. Fontaine, R., Aky, N., Courdier, R., Payet, D.: Vers une utilisation éco responsable des objets connectés par la mutualisation de leurs composants physiques: Une approche basée sur le concept d'artefact. In: Actes de la conférence nationale en intelligence artificielle CNIA - PFIA 2020, Angers, 2020. vol. 23, p. 38 (2020)
11. Gateau, B., Naudet, Y., Rykowski, J.: Ontology-based smart iot engine for personal comfort management. In: 2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP). pp. 35–40. IEEE, IEEE, Thessaloniki, Greece (2016)
12. IEEE: Middleware and application adaptation requirements and their support in pervasive computing. IEEE Computer Society, Los Alamitos, CA, USA (2003)
13. Jennings, N.R.: On agent-based software engineering. Artificial intelligence **117**(2), 277–296 (2000)
14. Kjær, K.E.: A survey of context-aware middleware. In: Proceedings of the 25th Conference on IASTED International Multi-Conference: Software Engineering. p. 148–155. SE'07, ACTA Press, USA (2007)
15. Kwan, J., Gangat, Y., Payet, D., Courdier, R.: A agentified use of the internet of things. In: Full Paper in 9th IEEE International Conference on Internet of Things(iThings 2016). IEEE CS (2016), http://hal.univ-reunion.fr/hal-01478263
16. Laleci, G.B., Dogac, A., Olduz, M., Tasyurt, I., Yuksel, M., Okcan, A.: Saphire: A multi-agent system for remote healthcare monitoring through computerized clinical guidelines. In: Annicchiarico, R., Cortés, U., Urdiales, C. (eds.) Agent Technology and e-Health. pp. 25–44. Birkhäuser Basel, Basel (2008). https://doi.org/10.1007/978 − 3 − 7643 − 8547 − 7$_3$, $https://doi.org/10.1007/978-3-7643-8547-7_3$
17. Maamar, Z., Faci, N., Boukadi, K., Ugljanin, E., Sellami, M., Baker, T., Angarita, R.: How to agentify the internet-of-things? In: 2018 12th International Conference on Research Challenges in Information Science (RCIS). pp. 1–6. IEEE (2018)
18. Mohamed, N., Al-Jaroodi, J., Jawhar, I.: Middleware for robotics: A survey. In: RAM. pp. 736–742. IEEE, Chengdu, China (2008)
19. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the a&a meta-model for multi-agent systems. Autonomous agents and multi-agent systems **17**(3), 432–456 (2008)
20. Pantoja, C.E., Soares, H.D., Viterbo, J., El Fallah-Seghrouchni, A.: An architecture for the development of ambient intelligence systems managed by embedded agents. In: SEKE. pp. 215–214 (2018)
21. Ramos, C., Marreiros, G., Santos, R., Freitas, C.F.: Smart offices and intelligent decision rooms. In: Nakashima, H., Aghajan, H., Augusto, J.C. (eds.) Handbook of Ambient Intelligence and Smart Environments, pp. 851–880. Springer US, Boston, MA (2010). https://doi.org/10.1007/978 − 0 − 387 − 93808 − 0$_3$2, $https://doi.org/10.1007/978-0-387-93808-0_32$
22. Russell, S.J.: Rationality and intelligence. Artificial Intelligence **1**(94), 57–77 (1997)
23. Seinturier, L., Merle, P., Rouvoy, R., Romero, D., Schiavoni, V., Stefani, J.B.: A Component-Based Middleware Platform for Reconfigurable Service-Oriented Architectures. Software: Practice and Experience **42**(5), 559–583 (May 2012). https://doi.org/10.1002/spe.1077, https://hal.inria.fr/inria-00567442
24. Weiser, M.: Some computer science issues in ubiquitous computing. Communications of the ACM **36**(7), 75–84 (1993)
25. Wooldridge, M.: Agent-based software engineering. IEE Proceedings-software **144**(1), 26–37 (1997)