An Applied Energy Management Approach in Intelligent Environments based on a Hybrid Agent Architecture

Thanos G. Stavropoulos [1, 2] and **Efstratios Kontopoulos** [3] and **Dimitris Sfounis** [1] and **Dimitris Vrakas** [1] and **Nick Bassiliades** [1,2] and **Ioannis Vlahavas**[1,2]

Abstract. This paper presents a framework for ambient sensing and managing a University building aimed at energy savings and user comfort. The system builds upon previous work, using a Semantic Web Service middleware for unifying the various heterogeneous sensor and actuator networks. Two applications are introduced in the framework, the Manager App and the Rule App. The latter incorporates a hybrid intelligent agent that enables both reactive and deliberate manipulation of the environment, based on user-configurable policies expressed in defeasible logic. The Manager App provides users with advanced control and renders the system sustainable. Specifically, it allows bypassing the policies and manually administrating the infrastructure, e.g. during exceptional or emergency cases. The framework is deployed and evaluated at a University course office, guaranteeing 20% daily energy savings on controlled devices, aggregated to 17% total per room savings.

1 INTRODUCTION

Ambient Intelligence (AmI) is one of the eminent concurrent and future computing paradigms, building upon Ubiquitous Computing (UbiComp) and Pervasive Computing (PerComp). According to these technological approaches, ambient and non-intrusive devices are scattered around the environment, in order to sense and act according to context. Therefore, such systems are also called Context-Aware and are mainly based on Semantic Web constructs (i.e. ontologies) for modelling context in a machine-interpretable way. So far, AmI systems were heavily based on Web Service technologies, such as WSDL, for publishing and exploiting universal, platform-agnostic APIs for the various devices that need to be used in such settings [9]. AmI state-of-the-art typically includes small scale applications targeting domains such as user comfort, office and home automation or even agriculture and healthcare (also known as Ambient Assisted Living – AAL). Such a categorization of AmI application domains can be found in [3].

This paper introduces a novel, hybrid agent architecture and applications as part of a holistic framework in the domain of energy savings and building management, targeted at a large-scale practical deployment. The Smart IHU project, designed and deployed at the International Hellenic University (IHU)[4], integrates different components on every level (hardware, middleware and application), in order to support this AmI environment. The hardware deployment consists of diverse heterogeneous and platform-dependent wireless networks of both sensors and actuators. After suitable drivers have been developed for these devices, the necessary abstractions for interfacing with them are provided by a Web Service middleware based on WSDL. On the semantic layer of the framework, WSDL descriptions have been further enriched with SAWSDL annotations to provide knowledge retrieval and reasoning mechanisms. While software applications on the upper layer of the framework handle user driven real-time and historic monitoring, intelligent agents are able to reason and act on the environment. To do so, the agents hold sets of energy-saving policies (in the form of knowledge bases) written by domain experts.

The paper is structured as follows: section 2 describes the framework's architecture, i.e. hardware and middleware integration, while the third section presents the proposed applications, namely Rule App and Manager App, followed by a description of the architecture of the proposed hybrid agent incorporated in Rule App. Section 5 presents an experimental evaluation of the complete framework, while the final section concludes with some final remarks and proposes directions for future work.

2 THE SMART IHU ARCHITECTURE

Figure 1 presents the proposed overall architecture for the Smart IHU system. Earlier variations of the system have been presented in previous work of ours, each of them serving different purposes. More specifically, [13] presents the underlying aWESoME middleware architecture, along with some simple client applications. Those applications, presented also in [14], allow a single administrator user to monitor historical and real-time sensor data and manually manage the infrastructure. In later work, the middleware was infused with semantic annotations, resulting in aWESoME-S, and the deployment of intelligent agents allowed the automatic manipulation of the infrastructure. Two agent types were interchangeably used within the framework, namely, a reactive agent, based on production rules, that rapidly reacts on environmental changes and a deliberative agent, based on defeasible logics [11], targeting "strategic" long-term energy-savings [12]. In this work, the semantically-infused middleware (aWESoME-S) is augmented with two novel applications, Manager App and Rule App, which allow both central and distributed operation. The latter, incorporates a hybrid agent that delivers both responsiveness and deliberation.

The hardware used within Smart IHU includes heterogeneous wireless sensor and actuator networks covering the whole Building A of the University. The sensors gather environmental measurements

[1] Aristotle University of Thessaloniki, Greece, email: {athstavr, dsfounis, dvrakas, vlahavas}@csd.auth.gr
[2] International Hellenic University, Greece, email: vlahavas@idu.edu.gr
[3] Information Technologies Institute, Greece, email: skontopo@iti.gr
[4] International Hellenic University homepage: `http://www.ihu.edu.gr/`

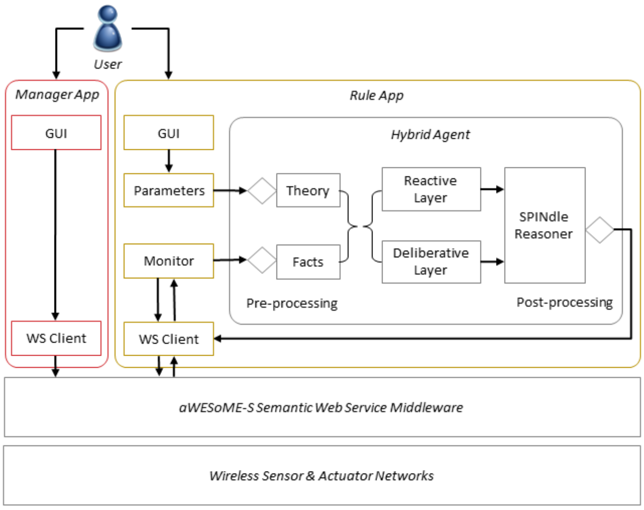

Figure 1. Smart IHU application architecture, showing underlying components and information flow within Manager App, Rule App and hybrid agent

(temperature, humidity, luminance and motion) from every room in the building as well as from outside. CO_2 concentration levels are also measured (Sensor Boards and Z-Wave devices) in selected classrooms. Another set of sensors/actuators gathers power consumption measurements from most electrical appliances, e.g. lights, PCs, printers and fan coils, also allowing to switch the devices on or off. Large-scale power consumption, e.g. building-level, the central cooling unit, coffee shop, PC labs and Auditorium, are collectively measured by Smart Clampers. These heterogeneous sensors, each providing a different communication interface and data format, are unified under the aWESoME-S middleware. aWESoME-S exposes universal SAWSDL endpoints that semantically describe sensor data retrieval and actuator functions. The annotations describe the nature of operation inputs and outputs, mostly according to the BOnSAI ontology [15], as described in [12]. Various clients exploit these endpoints for different purposes; e.g. a web application displays energy disaggregation, environmental conditions and waste metrics [5].

Note that the system may be distributed, but ultimately all commands have to go through two designated servers (one for the northern and one for the southern part of the building). Hence, there is a need to keep communication overhead to a minimum, in order to support various clients operating simultaneously.

3 THE SMART IHU APPLICATIONS

Some crucial requirements for the Smart IHU system are that it should be reliable, usable and available at all times. Meanwhile, the most critical role played by the system is automatic management of electrical appliances in the building, according to policies. While users might still adjust policies to their liking at all times, the system should provide the option of bypassing the rules and manually managing devices, e.g. at exceptional events or emergencies. Hence, two applications that run side-by-side, constitute the deployed Smart IHU system: (a) the Manager App, which allows manual management, and, (b) the Rule App, which automatically manages the infrastructure. Both applications can be configured to centrally target multiple or all rooms in IHU. This way, multiple application instances can be distributed to designated rooms. The coordination of those distributed software agents is also possible, and will be examined in future work.

The Smart IHU Manager App, first of all, retrieves relevant rooms, underlying appliances and appliance types from a central taxonomy and, consequently, allows the user to select an appliance from the hierarchy, and manually switch it on or off (via a Web Service client). In order to minimize average response time, the Manager App functionality has been kept to a minimum.

On the other hand, the Smart IHU Rule App is considerably more sophisticated. Similarly to the Manager App, it retrieves relevant rooms from a central taxonomy, but also retrieves a list of associated environmental and power sensors. The application incorporates a hybrid agent that monitors respective sensors and applies energy saving policies. While the agent's methodology for monitoring, maintaining and applying rules is presented in detail in the next section, from the Rule App perspective a user is able to parameterize those policies by providing allowed ranges or thresholds. He can also start and stop the agent altogether, configure the monitor loop intervals and individually activate rules.

That way, if a policy contradicts the user's preferences at a certain point, e.g. in an exceptional occasion, he can either re-adjust the rule parameters or temporarily deactivate the rule. Then, the Manager App can be used to switch back the appliances to the desired state. What sets Rule App apart from previous work [12], is that it translates the underlying rule logic to a user-friendly and rule-agnostic interface. While previous work lets experts freely author energy saving policies, this application operates on a higher level, already providing the policies and letting the user select them and parameterize them.

4 THE SMART IHU HYBRID AGENT

In this work we propose a *hybrid agent* that will handle the energy management within the building and will enforce energy-saving policies. A hybrid agent is concurrently capable of reactive and deliberative/proactive behaviors [17]. In recent work of ours [12], we highlighted the utility of this agent type for the specific application domain. This subsection presents our initial attempts towards implementing such an agent architecture.

The architecture of a hybrid agent typically integrates – amongst others – two distinct layers that deal respectively with the two types of behaviors (reactive and proactive). In essence, this principle of distinct layers is also adopted in our approach; as illustrated in Figure 1, the agent's architecture is horizontally layered, meaning that both layers accept sensory input and can produce action output.

In this work we adopt *defeasible logics* [11] as the knowledge representation formalism for expressing the agents' underlying logic. Defeasible logics are equipped with intuitive knowledge representation and advanced conflict resolution mechanisms, featuring the following:

- Three types of rules: *strict rules*, denoted by $A \rightarrow p$ (whenever the premises are indisputable, then so is the conclusion), *defeasible rules*, denoted by $A \Rightarrow p$ (can be defeated by contrary evidence) and *defeaters*, denoted by $A \leadsto p$ (do not actively support conclusions, but can only prevent deriving some of them).
- A binary *superiority relationship* that is used for resolving conflicts among defeasible rules and is denoted by $r_2 > r_1$, which declares that rule r_2 is superior to r_1 and eventually prevails.
- Sets of conflicting literals, each of which allow only one at most literal of the set to be derived each time.

[5] The Smart IHU Web Application: `http://smart.ihu.edu.gr`

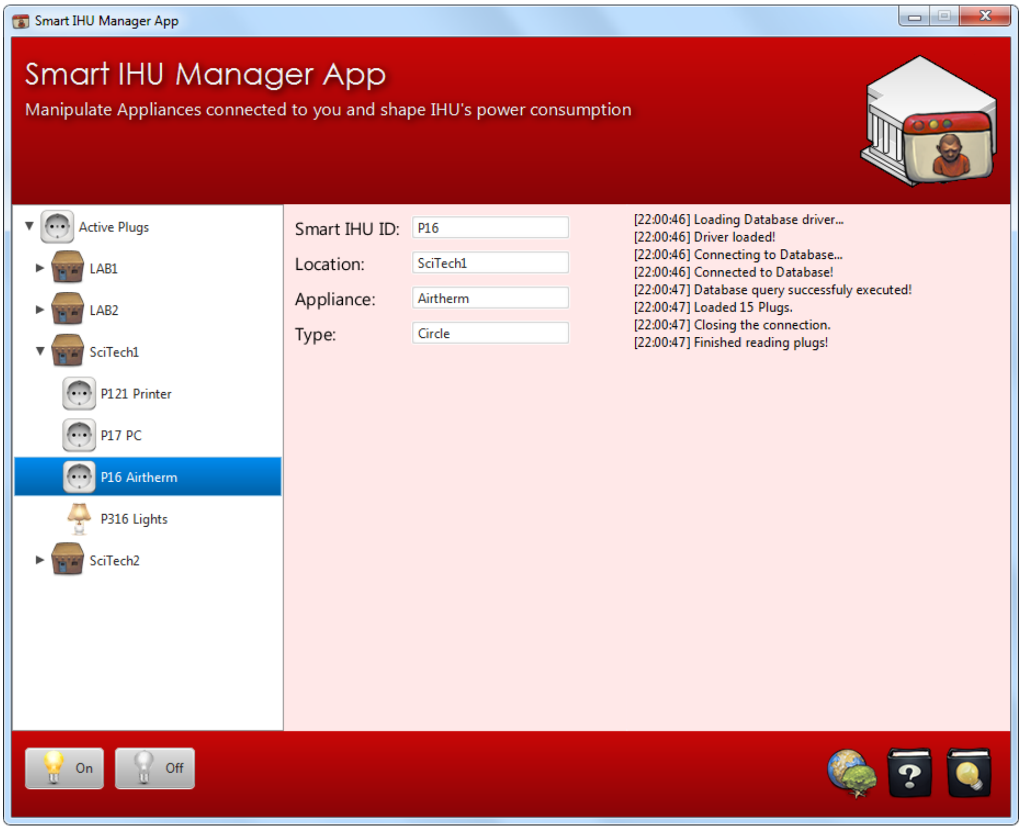

Figure 2. The Manager App GUI, allowing users to manually manage each room's controlled devices

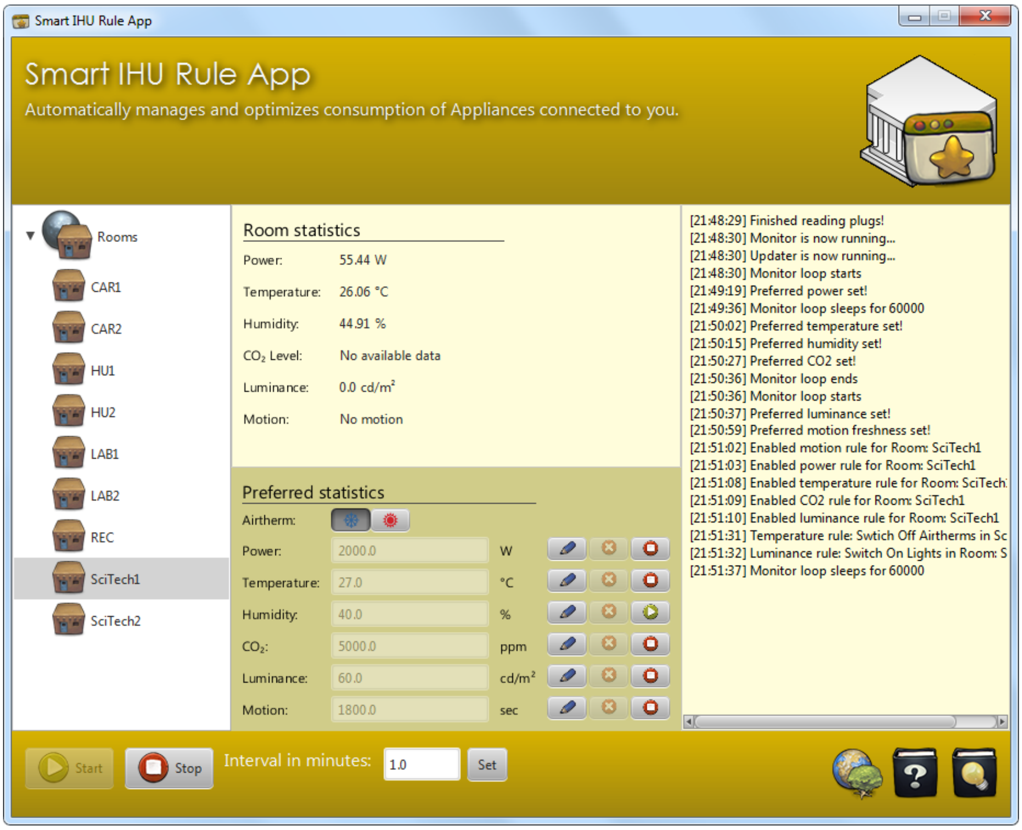

Figure 3. The Rule App GUI, continuously monitoring parameters, allowing policy configuration and selection

Specifically in this work we conceive defeasible logics as a knowledge representation paradigm that inherently integrates both the reactive (in the form of strict rules and facts) and the deliberative (in the form of facts, defeasible rules, defeaters, conflicting literals and superiority relationships) aspects. Consequently, the two layers practically handle different subsets of the agent's knowledge base. The latter, which represents the agent's policies in the form of rules, is then submitted to the rule engine (SPINdle – see next paragraph) that executes the rule set and derives conclusions. This "unified layers" approach removes the burden of having a mediator module – typical in hybrid agent architectures – that would be responsible for determining which of the suggestions by the different layers has to be eventually adopted.

For the reasoning process, the proposed agent incorporates an instance of *SPINdle* [10], a Java-based state-of-the-art defeasible reasoning engine. Although SPINdle currently lacks a theory grounding mechanism (i.e. no variables can be used in the predicates), which would be greatly beneficial for the purposes of this work, the engine's advantages (fast, reliable, highly integrated and easily-deployable) constitute the incentive for preferring the specific reasoner. The lack of grounding mechanism is solved in our approach through the Pre-processing module (see Figure 1), which replaces atoms containing arguments and variables with appropriately composed sets of synthetic predicates. For example, atom `switchOn(cooler)` becomes `switchOn_cooler`, while atom `switchOn(X)` is replaced by a set of synthetic predicates `switchOn_device_1`, `switchOn_device_2`, ..., `switchOn_device_n` for all the *n* registered devices. In case SPINdle supports first-order theories in the future, this pre-processing phase will not be necessary any more.

The rest of this section features a simplified example that demonstrates the deployment of the policies rule base and the distinction between the reactive and the deliberative aspects of the agent. Suppose that the following example involves a hybrid agent monitoring a specific room 'a1'. Its reactive policies subset is:

```
r11: motion(a1), ¬isOn(a1,light) →
    switchOn(a1,light)
r12: ¬motion(a1), isOn(a1,light) →
    switchOff(a1,light)
```

On the other hand, the agent's deliberative policies would be more sophisticated, featuring the following rules:

```
r21: temp(a1,X), X > 25 ⇒ switchOn(a1,cooler)
r22: temp(a1,X), X < 25 ⇒
    switchOff(a1,cooler)
r23: consumption(a1,X), X > 2000 ⇒
    savingMode(a1)
r24: time(a1,X), X > 2300 ⇒ savingMode(a1)
r25: savingMode(a1) ⇒ switchOff(a1,cooler)
r26: alert(a1) ⇒ ¬savingMode(a1)
r27: isOn(a1,X) ⇝ ¬switchOn(a1,X)
r28: isOff(a1,X) ⇝ ¬switchOff(a1,X)
C = {switchOn(X,Y), switchOff(X,Y)}
r25 > r21
r26 > r23
r26 > r24
```

Notice that, as already mentioned previously, there are no strict rules in the latter rule set. Furthermore, the defeaters (r_{27} and r_{28}) ensure that the device actuators in room a1 will not attempt to turn on an already operating device and, conversely, will not attempt to turn off a device that has already been switched off. In turn, the conflicting literal set C makes sure that pairs of rules attempting to respectively switch on and off devices will be considered as conflicting and will be resolved via the superiority relationships appended in the end of the rule set.

According to the above set of rules, whenever there is motion within the room, the lights are switched on (rules r_{11} and r_{12}) and the same goes for the cooler when the temperature is above 25^{o}C (rules r_{21} and r_{22}). On the other hand, the agent enters a "saving mode" every time the total consumption within the room goes above a specified threshold or/and during late night hours (rules r_{23} and r_{24}). The "saving mode" status is superior to the normal device operation ($r_{25} > r_{21}$); however, the case of an alert prevails over all the previous conditions (rule r_{26} and superiority relationships $r_{26} > r_{23}$ and $r_{26} > r_{24}$).

5 EXPERIMENTAL EVALUATION

The system was deployed at the IHU premises and was put to use. Due to the large scale of the experiment, targeting the whole building consumption at once derives results hard to assess, as discusses in previous work of ours [12]. Specifically, some critical factors are not yet dealt with, such as cooking appliances at the coffee shop, central cooling and heating. Therefore, the proposed framework was experimentally evaluated in a specific office, namely the course office of the Science and Technology School. In view of our future improvements, we take into account that other offices within the building present similar behaviour: same environmental conditions, energy demands and equipment per person.

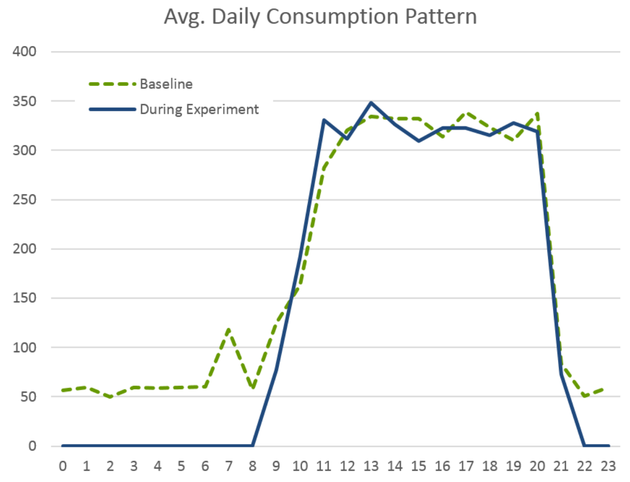

Figure 4. Average daily consumption as measured during a) the experimental evaluation and b) two weeks ago as a baseline. The y-axis shows consumption in Watts (W), the x-axis shows hour of the day.

During the experiment, office employees were able to use both the Manager App and the Rule App from a dedicated laptop at all times. They could freely set thresholds, activate or deactivate policies or the Agent altogether. The experiment run between 08-May-2014 and 21-May-2014, while the interval between 24-Apr-2014 and 07-May-2014 was used for measuring the baseline consumption. Average daily consumption during both intervals is shown in Figure 4 per hour of the day. The peak observed at 6am to 8am is due to the cleaning crew leaving all lights on when interleaving rooms.

Measurements during the experiment are summarized in Figure 5. The table shows "On" Consumption (average consumption while the device is on) for each controlled device, i.e. lights, printers and fan coils, indicatively. Baseline measurements and savings during the experiment are also listed for each of the devices, their total and room total. Based on past consumption baseline, a 20.07% reduction on daily energy consumption of controlled devices was observed. This amount of savings is aggregated to 17.30%, including the rest of the devices in the room (which includes mostly PCs). Meanwhile, several observations were made during the experiment:

- The Manager App was almost not used at all, as no emergencies, special occasions or malfunctions occured.
- The motion rule manipulating the printer, eliminates printer standby when the office is empty (e.g. visible at 0am - 6am). This results in significant savings, i.e. 18.85% on controlled devices and 14.59% on room total.
- The motion rule manages to eliminate the peak during 6am - 8am, by switching lights off whenever the cleaning crew leaves a room.
- The temperature and power rules did manage to periodically deactivate fan coils. Regardless, fan coil consumption is insignificant and has no impact on neither consumption nor savings (0%).
- The luminance rule also had no impact on savings since the employees do need the lights on during work. Thus, they had set a high preferred threshold. Consequently, light manipulation is entirely due to motion, increasing savings by 1.22% on controlled devices or 0.95% on total room consumption.
- During office hours all equipment is on, achieving no savings (i.e. 9am - 10pm).

The experiment highlights two future directions for improvement. First, the fan coils may consume an insignificant amount of energy, but require cool water from a central cooling unit. This unit is responsible for a massive percentage of energy (ca. 20kW, 50% of the building's consumption). In order to manage to switch central cooling off, we plan to timely coordinate all fan coils in the building to consume no cool water.

Secondly, to improve on luminance preferences, we plan to intersect more light cables to provide scalable lighting. Specifically, each actuator is now placed to control all lights within a room. Intersecting each light in a room and changing the rules accordingly will provide the users with scalable lighting according to preference.

6 STATE OF THE ART

Current state-of-the-art is examined in many directions such as rule-based context-aware systems, AmI middleware and defeasible logics applications in AmI. As far as rule-based smart environments are concerned, one approach introduced in [5] is a meta-language defined over JESS, to syntactically enhance the rule authoring process in ambient applications. However, this additional syntactic layer, named the Event-Control-Action (ECA) model, is far less flexible and extensible than defeasible logics used in the proposed framework. Other similar approaches include SESAME-S [6], an all-in-a-box smart home prototype that uses ontologies and JESS reasoning to enforce rules. The main drawback is the lack of conflict resolution and scalability, while our work targets those aspects with defeasible logic. The Case-Based Reasoning Agent (CBR), [18] also employs Web Services, sensor middleware and a domain ontology for knowledge representation. However, it does not present a thorough methodology on achieving and measuring energy savings. Additionally, services are used to improve performance over the Cloud only, logics are not investigated for expressiveness and maintainability, while our set of devices is richer.

Regarding middleware, most existing paradigms use more complex and, thus, less interoperable top-down standards, as in [16]. Similarly, custom languages are used in Hydra [4], a middleware which enables device integration in AAL, and the AmIi [7] middleware, which maps description languages to an internal one. On the contrary, the aWESoME-S middleware used in this work [12] exposes SAWSDL endpoints, which are fully interoperable across the web, straightforward but also expressive enough for matching and composition [8]. It also integrates a sufficient set of devices for the proposed applications.

Defeasible logic has been already applied to an AmI setting in [2], where context knowledge is distributed across agents. Defeasible logic is used both as a basis for context knowledge representation and for resolving potential conflicts emerging from the information exchange between agents. Furthermore, the authors have introduced the *Contextual Defeasible Logic (CDL)* [18] extension [1]; a reasoning model to address challenges posed by the dynamic nature of AmI environments, by using the concepts of context, mappings and contextual preferences. These approaches are similar to the one presented in this paper, in the sense that they both target AmI environments with defeasible logics in real-world, practical applications. On the other hand, the approach proposed here, poses many differences. The Smart IHU agent, logic and middleware are implemented, deployed and practically used in a real university environment, while [2] focuses on the reasoning model only. Additionally, defeasible logic is used for handling conflicts between agent exchange in [2], rather than intuitive policy and knowledge representation as in our work. Meanwhile, [2] addresses the AAL domain with a distributed approach, while this work proposed a centralized system for the energy-management domain.

7 CONCLUSIONS AND FUTURE WORK

This work presents a framework for energy management, both manual and automatic, in intelligent environments. The framework is applied in a Smart University deployment of wireless sensors and actuators and an existing Semantic Web Service middleware to provide uniform access. Two applications on top of the middleware aid in automatic and manual management, respectively. The Rule App, incorporates a Hybrid intelligent Agent, which demonstrates both reactive and deliberative behaviour, each handling policies of different priority. The policies are expressed in human-intuitive and easier to maintain defeasible logics, while parameters can be changed via a user-friendly GUI. The Manager App facilitates the sustainability and acceptability of the system, since it can bypass rules and manually manage the infrastructure in cases of emergency or exceptional events. The system was experimentally evaluated during two weeks of operation at a designated University course office demonstrating around 20% of daily energy savings on controlled devices, or 17% on total per room savings.

Future steps include the systematic deployment of the applications across more offices within the University building, in order to measure a larger scale impact. Additionally, we plan to measure the effects on the central cooling system during summer months. While each fan coil in offices and rooms consumes almost no energy daily, the central cooling unit that distributes cool water to fan coils holds around 50% of the University's total consumption. Hence, we plan to investigate the impact on cooling after enforcing timely, policy-driven fan coil operation. For that purpose, a multi-agent architecture and coordination schemes are being investigated.

Appliance	"On" Consumption (W)	Hours Off (h)	Baseline Avg. Daily Power (W)	Baseline Avg. Daily Energy (Wh)	Experiment Avg. Daily Power (W)	Experiment Avg. Daily Energy (Wh)	Savings	Savings/ Total Savings (%)	Savings/ Baseline Total (%)	Savings/Baseline Room Total (%)
Lights	185.42	0.22	80.89	1941.45	79.19	1900.63	40.82	5.46%	1.22%	0.95%
Fan Coil	0.05	3.20	0.02	1400.94	0.01	0.24	0.16	0.02%	0.00%	0.00%
Printer	64.91	9.71	58.37	1400.94	32.12	770.88	630.05	84.34%	18.85%	14.59%
Total	n/a	n/a	139.28	3342.79	111.32	2671.75	671.03	89.82%	20.07%	15.54%
Room Total	n/a	n/a	179.91	4317.85	148.78	3570.79	747.05	100.00%	n/a	17.30%

Figure 5. Experimental evaluation measured savings per controlled device, total and room total.

ACKNOWLEDGEMENTS

The authors would like to thank Konstantinos Gottis, John Argyriou, Theodoros Tsompanidis and George Pilikidis for their work on various parts of the Smart IHU system.

The Smart IHU project is funded by Operational Program Education and Lifelong Learning, OPS200056 (International Hellenic University, Thessaloniki, Greece).

REFERENCES

[1] Antonis Bikakis and Grigoris Antoniou, 'Defeasible contextual reasoning with arguments in ambient intelligence', *Knowledge and Data Engineering, IEEE Transactions on*, **22**(11), 1492–1506, (2010).

[2] Antonis Bikakis, Grigoris Antoniou, and Panayiotis Hasapis, 'Strategies for contextual reasoning with conflicts in ambient intelligence', *Knowledge and Information Systems*, **27**(1), 45–84, (2011).

[3] Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula, 'Ambient intelligence: Technologies, applications, and opportunities', *Pervasive and Mobile Computing*, **5**(4), 277–298, (2009).

[4] Markus Eisenhauer, Peter Rosengren, and Pablo Antolin, 'A development platform for integrating wireless devices and sensors into ambient intelligence systems', in *Sensor, Mesh and Ad Hoc Communications and Networks Workshops, 2009. SECON Workshops' 09. 6th Annual IEEE Communications Society Conference on*, pp. 1–3. IEEE, (2009).

[5] Richard Etter, Patricia Dockhorn Costa, and Tom Broens, 'A rule-based approach towards context-aware user notification services', in *Pervasive Services, 2006 ACS/IEEE International Conference on*, pp. 281–284. IEEE, (2006).

[6] Anna Fensel, Slobodanka Tomic, Vikash Kumar, Milan Stefanovic, Sergey V Aleshin, and Dmitry O Novikov, 'Sesame-s: Semantic smart home system for energy efficiency', *Informatik-Spektrum*, **36**(1), 46–57, (2013).

[7] Nikolaos Georgantas, Valerie Issarny, Sonia Ben Mokhtar, Yerom-David Bromberg, Sebastien Bianco, Graham Thomson, Pierre-Guillaume Raverdy, Aitor Urbieta, and Roberto Speicys Cardoso, 'Middleware architecture for ambient intelligence in the networked home', in *Handbook of Ambient Intelligence and Smart Environments*, 1139–1169, Springer, (2010).

[8] Guilherme C Hobold and Frank Siqueira, 'Discovery of semantic web services compositions based on sawsdl annotations', in *Web Services (ICWS), 2012 IEEE 19th International Conference on*, pp. 280–287. IEEE, (2012).

[9] Valerie Issarny, Mauro Caporuscio, and Nikolaos Georgantas, 'A perspective on the future of middleware-based software engineering', in *2007 Future of Software Engineering*, pp. 244–258. IEEE Computer Society, (2007).

[10] Ho-Pun Lam and Guido Governatori, 'The making of spindle', in *Rule Interchange and Applications*, 315–322, Springer, (2009).

[11] Donald Nute, 'Defeasible logic', *Handbook of logic in artificial intelligence and logic programming*, **3**, 353–395, (1994).

[12] T. G. Stavropoulos, E. Kontopoulos, N. Bassiliades, J. Argyriou, A. Bikakis, D. Vrakas, and I. Vlahavas, 'Rule-based approaches for energy savings in an ambient intelligence environment', *Journal of Pervasive and Mobile Computing*, to appear.

[13] Thanos G Stavropoulos, Konstantinos Gottis, Dimitris Vrakas, and Ioannis Vlahavas, 'awesome: A web service middleware for ambient intelligence', *Expert Systems with Applications*, **40**(11), 4380–4392, (2013).

[14] Thanos G Stavropoulos, Dimitris Vrakas, Alexandros Arvanitidis, and Ioannis Vlahavas, 'A system for energy savings in an ambient intelligence environment', in *Information and Communication on Technology for the Fight against Global Warming*, 102–109, Springer, (2011).

[15] Thanos G Stavropoulos, Dimitris Vrakas, Danai Vlachava, and Nick Bassiliades, 'Bonsai: a smart building ontology for ambient intelligence', in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, p. 30. ACM, (2012).

[16] Graham Thomson, Sébastien Bianco, Sonia Ben Mokhtar, Nikolaos Georgantas, and Valérie Issarny, 'Amigo aware services', in *Constructing Ambient Intelligence*, 385–390, Springer, (2008).

[17] Michael Wooldridge, *An introduction to multiagent systems*, John Wiley & Sons, 2009.

[18] Sheng-Yuan Yang, 'Developing an energy-saving and case-based reasoning information agent with web service and ontology techniques', *Expert Systems with Applications*, **40**(9), 3351–3369, (2013).