An Energy Aware Context Model for Green IT Service Centers

Ioan Salomie[1], Tudor Cioara[1], Ionut Anghel[1],
Daniel Moldovan[1], Georgiana Copil[1] and Pierluigi Plebani[2]

[1]Technical University of Cluj-Napoca, Computer Science Department,
Cluj-Napoca, Romania
{Ioan.Salomie, Tudor.Cioara, Ionut.Anghel}cs.utcluj.ro

[2] Politecnico di Milano, Computer Engineering,
Milano, Italy
Plebani@elet.polimi.it

Abstract. In this paper we propose the development of an Energy Aware Context Model for representing the service centre energy/performance related data in a uniform and machine interpretable manner. The model is instantiated at run-time with the service center energy/performance data collected by monitoring tools. Energy awareness is achieved by using reasoning processes on the model instance ontology representation to determine if the service center Green and Key Performance Indicators (GPIs/KPIs) are fulfilled in the current context. If the predefined GPIs/KPIs are not fulfilled, the model is used as primary resource to generate run-time adaptation plans that should be executed to increase the service center's greenness level.

Keywords: Service Center, Context Model, Energy Awareness, Reinforcement Learning

1 Introduction and Related Work

Over the last years the energy efficiency management of IT processes, systems and service centers has emerged as one of the most critical environmental challenges to be dealt with. Since computing demand and energy costs are continuously growing, energy consumption of IT systems and service centers is expected to become a priority in the future years [1].

The GAMES (Green Active Management of Energy in IT Service Centers) EU FP7 research project [2] aims at developing a set of innovative methodologies, metrics, services and tools for the active management of energy efficiency of IT service centers. The GAMES vision is to create a new generation of Green and Energy Aware IT service centers by defining and implementing management actions in both design time and run-time for increasing energy efficiency. The problem of service centre run-time energy efficiency in the GAMES project is approached by dynamically finding and executing Dynamic Power Management (DPM) and Consolidation based adaptation actions targeting the identification of the over

provisioned resources with the goal of putting them in low power states. To determine run-time adaptation decisions, the following MAPE (Monitoring, Analysis, Planning and Execution) steps are taken: (i) the current service center energy/performance data is captured using monitoring tools, (ii) using the collected data, the current values for GPIs (Green Performance Indicators) and KPIs (Key Performance Indicators) are evaluated and compared against their predefined values (iii) if the GPIs and KPIs are fulfilled no action is taken; otherwise a plan of adaptation actions is determined and executed to enforce the GPIs/KPIs predefined values.

The service center energy/performance data collected from various sources (such as sensors, monitoring devices, software applications, etc.) is represented in heterogeneous formats that are difficult to interpret and analyze. There is an evident need for providing an integrated, uniform and semantically enhanced energy/performance data representation model that can be automatically processed and interpreted at run-time. To solve these problems we have developed an Energy Aware Context Model (EACM). The proposed EACM model is constructed by mapping our RAP (Resources, Actions and Policies sets) generic context model [16] onto the service centre energy efficiency domain. An EACM instance representing a service centre snapshot is created by instantiating at run-time the EACM model elements with the service center energy/performance data collected using monitoring tools. To ensure the energy awareness, the EACM model instance, implemented as ontology is analyzed/processed at run-time by using reasoning processes, to determine if the GPIs/KPIs are fulfilled for the current service center snapshot and to generate/execute adaptation action plans if these indicators are not fulfilled.

The state of the art literature contains many references regarding context modeling, but none of them (as of our knowledge) approaches the energy efficiency problem by considering the energy consumption as a relevant context feature. Also, there are no approaches for context modeling of Green IT service centers.

The most important problems regarding context information acquisition refer to identifying the features of the system execution context [3], [20], and defining models for capturing features' specific data [4]. In the domain literature ([5], [6] and [7]), several characteristics that may define the context are considered, such as spatiotemporal (time and location), ambient, facility (the system devices and their capabilities), system user interaction, system internal events, system life cycle, etc. Our paper takes this work one step further and introduces the energy consumption and the resource usage as an important characteristic of the modeled context.

Regarding the context representation, generic models that aim at accurately describing the context in a programmatic manner are proposed [19]. In [8] the use of key-value models to represent the set of context features and their associated values is proposed. Markup models [9] and object oriented models [10] are also proposed to structure and represent the context information. The main disadvantage of these approaches is their high degree of inflexibility and lack of semantics. Alternatively, the use of ontologies to model the context data, where context features are represented as ontological concepts and instantiated with run-time captured values are more and more used [11]. We took advantage of the semantic-oriented and reasoning features of the ontologies and developed an ontology based representation of the EACM model.

For context analyzing, models and techniques aiming at determining and evaluating the context changes are proposed. These models are strongly correlated

with the context representation model. In [12] fuzzy Petri nets are used to describe context changing rules. Context analyzing models based on reasoning and learning about context information are proposed in [13], [14] and [15] where context change rules are described using natural language or first order logic and evaluated using reasoning engines. Our paper uses reasoning algorithms to evaluate and analyze the run-time changes targeting the energy awareness.

The rest of this paper is structured as follows: Section 2 presents the EACM model, Section 3 shows how energy awareness is enacted, Section 4 describes a case study and evaluation results, while Section 5 concludes the paper.

2 EACM – The Energy Aware Context Model

This section introduces the EACM model highlighting the main concepts and relations defined and used to represent the service center energy / performance related data. The EACM model is constructed by mapping the RAP context model [16] onto the service centre energy efficiency domain. In the RAP model the context data is represented as triple <R, A, P> where R is the set of context resources, A is the set of context adaptation actions and P is the set of context policies. *Context resources (R)* represent the physical or virtual entities that generate and / or process context data. *Context actions (A)* represent a set of possible adaptation actions that have to be executed to enforce a predefined set of conditions for a given context situation. *Context policies (P)* are used to define a set of rules that must hold in the context, being used for controlling the interactions within the context. Fig. 1. shows the service centre energy efficiency domain specific concepts (highlighted in blue) and their classification into the RAP context model main sets (highlighted in red).

Context Resources (R). Three types of context resources that generate/collect context data were identified in service centers: (1) Service Centre IT Facility Context Resources (Facility Resource for short), (2) Service Centre IT Computing Context Resources (Computing Resource for short) and (3) Business Context Resources (Business Resource for short). *Facility Resources* are physical or virtual entities which provide or enforce the service centre ambient properties. A Facility Resource is characterized by the ambient property data type that resource can capture or modify. Passive Resources capture and store service centre ambient data, while Active Resources execute adaptation actions to modify the service centre ambient properties. For example, a temperature value can be the property for a temperature sensor resource (non-modifiable property) as well as for an air cooling resource (this time a modifiable property). *Computing Resources* are physical or virtual entities which supply context data related to the actual workload and performance capabilities of the service center. A Computing Resource can be also defined as a resource which consumes energy as a result of executing a specific workload. In our model we are interested to represent only those service centre computing resources that allow executing of Dynamic Power Management (DPM) actions aiming at setting a computing resource into different power states, according to its current workload. A Computing Resource is characterized by the list of energy consuming states property.

For example, an Intel Core i7 860@2.8Ghz processor has 12 P-states varying from 1197Mhz (40%) up to 2926Mhz (100%). The Computing Resources are classified as Simple and Complex (see Fig. 1). A Simple Computing Resource provides only one atomic performance property throughout its lifecycle. For example, CPU energy-related performance is characterized only by its frequency value. A Complex Computing Resource is composed from a set of Simple Resources and is characterized by a set of performance properties. For example the energy-related performance of a server is expressed by means of its component's energy-related performance properties such as the spindle speed for HDD or the clock rate for CPU. A *Business Resource* is a virtual entity which provides information about the QoS requirements of the executed application.

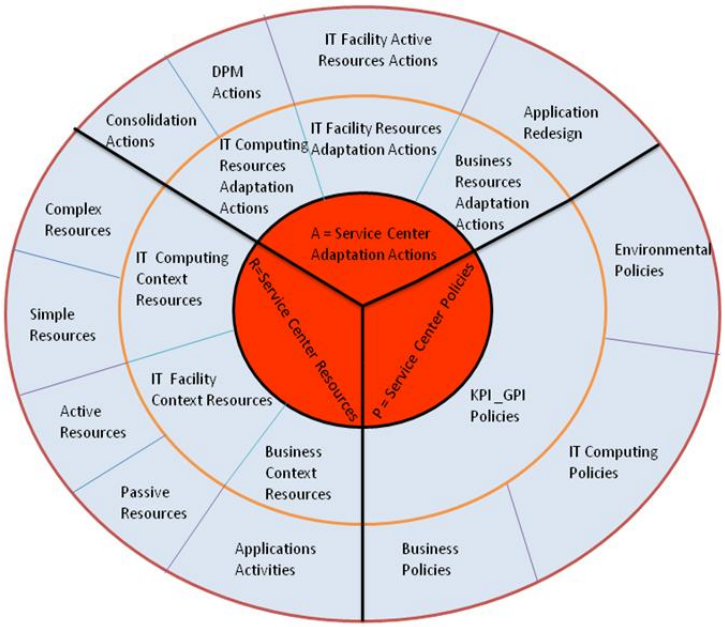

Fig. 1. EACM model elements obtained by mapping RAP model onto service center energy efficiency domain.

Context Actions (A). Three types of service centre adaptation actions are identified: (1) IT Computing Resources Adaptation Actions, (2) IT Facility Resources Adaptation Actions and (3) Application Adaptation Actions. *IT Facility Resources Adaptation Actions* (e.g. adjust the room temperature or start the CRAC) are enforced through the Active Resources. *IT Computing Resources Adaptation Actions* are executed to enforce the set of predefined GPI and KPI indicators on the Computing Resources. We have defined two types of IT Computing Resources Adaptation Actions: Consolidation Actions and DPM Actions. Consolidation Actions aim at identifying the Computing Resources for which the workload is inefficiently distributed from the energy efficiency point of view and balancing workload distribution in an energy efficient manner. DPM Actions aim at determining the over

provisioned resources with the goal of putting them into low power states. *Application Adaptation Actions* should be executed during design-time on the service centre applications activities (such as application redesign for energy efficiency).

Context Policies (P). The constraints regarding the service centre energy / performance are modeled through a predefined set of GPI and KPI related policies. We have identified and modeled three categories of GPI and KPI policies (see Fig. 1): (1) Environmental Policies, imposing restrictions about the service centre ambient conditions, (2) IT Computing Policies, describing the energy/performance characteristics of the service centre that and (3) Business Policies, describing the rules imposed by the custom business for the application execution. The Context Policies are described using the XML based policy description model proposed by us in [18].

Energy Aware Context Model Elements Relations. To model the interactions between the EACM model elements we have defined three types of relations: (1) proper subset relations, (2) trans-set 1 to N relations and (3) trans-set 1 to 1 relations. The *proper subset relations* reflect the hierarchical relations between the EACM model elements. For example, the Context Resources set is populated with service centre resources classified in three main proper subsets: Computing Resources, Facility Resources and Business Resources. The *trans-set 1 to N relations* model the interactions between an element of an ECAM model set and a subset of elements of the model. For example, this type of relation is used to model and represent the interactions between the energy aware run-time adaptation action and the service center resources on which it is enforced. The *trans-set 1 to 1 relations* model the interactions between two EACM model elements part of different sets. For example, in our model a policy (part of the Context Policy set) may have attached a default adaptation action (part of the IT Computing Adaptation Actions set) that has to be executed when the policy is broken.

3 Enacting Energy Awareness

To assure energy awareness, the service center context situation (snapshot) is represented in a programmatic manner using the EACM model instance ontology implementation (see Fig. 2). The EACM model instance is processed and interpreted at run-time by means of reasoning to: (1) evaluate if the GPIs/KPIs context policies are fulfilled for the current service centre context situation and (2) generate/execute adaptation action plans if the GPIs/KPIs policies are not fulfilled.

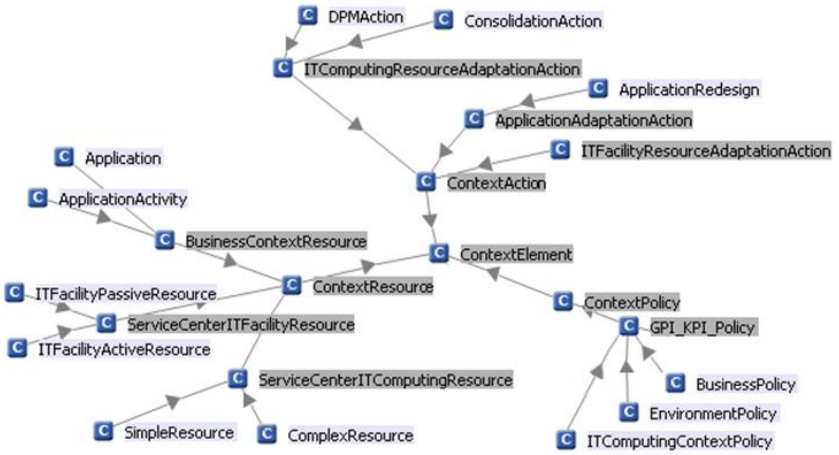

Fig. 2. EACM model elements implemented as ontological classes.

To *evaluate the GPI/KPI policies*, reasoning rules are used. The policies are converted into reasoning rules and automatically evaluated (without human intervention) by means of a reasoning engine, using the EACM model instance ontology representation. Fig. 3 shows the evaluation of GPIs/KPIs context policies using as an example a policy describing the accepted performance/workload values for a server and its SWRL (Semantic Web Rule Language) rules representation.

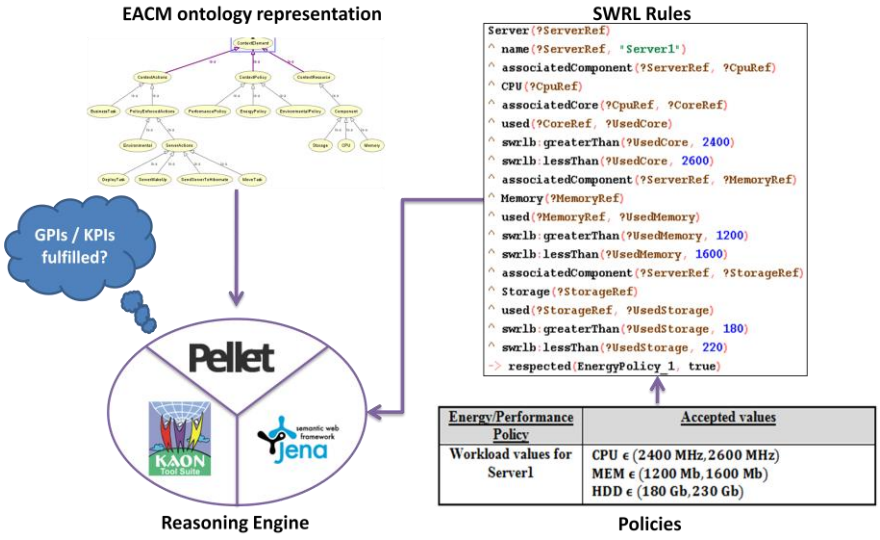

Fig 3. The GPI/KPI policy evaluation.

To measure the degree of fulfilling the set of GPIs/KPIs related policies we have defined the concept of service centre context situation entropy (E_S) and its associated

threshold (T_E). The entropy is an indicator that measures the level of compliance to the service center specific energy-saving requirements (i.e. the greenness level) [21]. If the evaluated context situation entropy is below a predefined threshold T_E, then all the GPIs/KPIs policies are fulfilled and adaptation is not required. Otherwise, adaptation actions must be executed to enforce the broken GPIs/KPIs policies and to bring the entropy below T_E. The EACM model instance entropy is computed as in relation (1) where: (i) pw_i is the weight of the GPI/KPI policy i and represents the importance of the policy in the service centre context, (ii) rw_{ij} is the weight of the service centre context resource i in the policy j and reflects the service centre resource importance for that policy and (iii) v_{ij} is the deviation between the recorded value for service centre resource j and the accepted value defined by policy i.

$$E_S = \sum pw_i \sum rw_{ij} * v_{ij} \quad (1)$$

Taking into account the toleration to changes, we have defined two types of entropy thresholds: *a restrictive threshold* and *a relaxed threshold*. For the first case, we define the threshold at the lowest possible entropy value ($T_E = 0$). Whenever a GPI/KPI policy imposed restriction is broken, the adaptation process is triggered. In the second case, for each GPI/KPI policy we define an accepted entropy contribution value E_i and compute the entropy threshold T_E (relation 2). The broken GPIs/KPIs policies are tolerated if their entropy contribution is lower than the accepted value.

$$T_E = \sum pw_i * v_{ij} * (100 + E_i \,\%\, 100) \quad \text{where} \quad E_i = pw_i \sum ri_{ij} * v_{ij} \quad (2)$$

To *generate and execute adaptation action plans*, we identify first the previously encountered similar service center context situations in which the same GPIs/KPIs context policies were broken. If such a similar equivalent situation is found, the same action plan is selected and executed (see Fig 4).

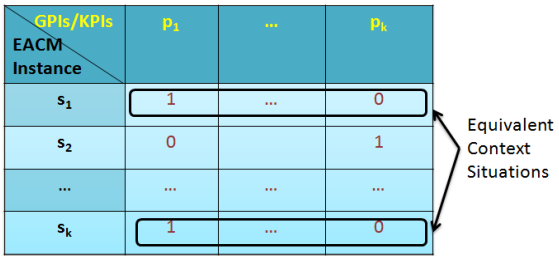

Fig 4. Equivalent service center context situations.

Otherwise, a new sequence of adaptation actions is generated using a reinforcement learning approach (what / if analysis). The learning process considers all possible service center context situations (represented as EACM instances) and builds a decision tree by simulating the execution of all available adaptation actions for each situation (see Fig 5). Each tree node stores a service center context situation and its calculated entropy value. A tree path between two nodes S1 and S2 defines a

sequence of adaptation actions which, executed in S1 context situation, generates the new service center context situation stored in node S2. The minimum entropy path in the reinforcement learning decision tree represents the best sequence of adaptation actions that when executed, will bring the service center in a context state in which all GPIs/KPIs context policies are fulfilled.

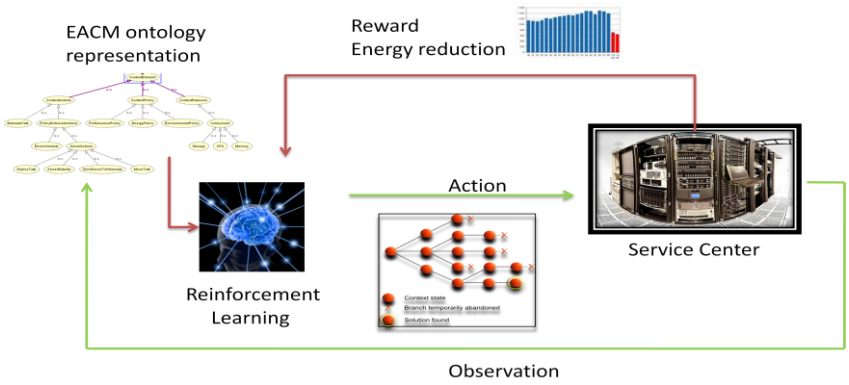

Fig 5. Reinforcement learning based adaptation action plans generation.

The learning process may generate different type of results discussed below.

Case 1: The algorithm finds only a possible service center context situation with an entropy value lower than the defined threshold. The sequence of actions that lead to this context situation is selected and the search process is stopped.

Case 2: The current service center context situation entropy is higher than the threshold, but smaller than the minimum entropy determined so far. The minimum entropy is replaced with the new entropy and the search process is continued.

Case 3: The current entropy is higher than both the threshold and the minimum entropy; the reinforcement learning algorithm continues the search process. If at a certain moment, all exercised paths of the decision tree are cycles, the algorithm stops and chooses the path leading to a state with the minimum entropy.

4 Case Study

In this section the *energy awareness capabilities* of the proposed EACM model and *the scalability* of its ontology representation are discussed and evaluated.

To **evaluate EACM model energy awareness capabilities,** we used it to manage a small service center with the following configuration: (i) a server cluster composed from three physical servers on which virtualized applications, annotated with Quality-of-Service (QoS) requirements, are executed, (ii) an external shared storage server and (iii) a set of sensors and facilities interconnected through a sensor network which control the service centre environment. Fig. 6 presents the test case service centre infrastructure together with the set of defined GPIs/KPIs policies.

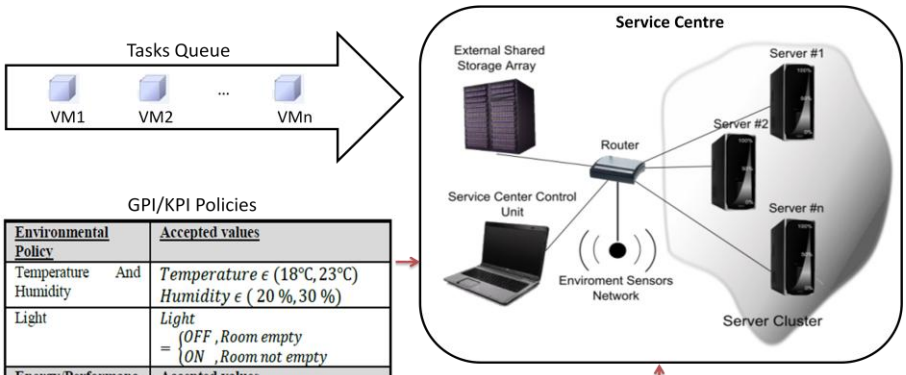

Fig 6. Test case service centre infrastructure.

To determine and analyze the time necessary for *evaluating the GPIs/KPIs context policies*, we have generated two categories of service center context situations: (i) situations in which the number of GPIs/KPIs broken policies is gradually increasing (each broken policy having similar complexity) and (ii) situations in which GPIs/KPIs with increasing complexity are broken. The complexity is measured in terms of the number of atoms in the antecedent of the policy corresponding SWRL rule. For the first context situations category, the test started from three broken GPIs/KPIs policies and continued by gradually increasing this number up to 30 broken policies. For the second category of context situations, starting from one broken GPI/KPI policy with 4 atoms the number of atoms was grown up to 34. The results show (Fig. 7b) that for less than 20 atoms the evaluation time is within reasonable limits (less than 0.15sec.). When the complexity of the policy increases above 20 atoms the evaluation time grows exponentially. Reasonable time results (Fig. 7a) were also obtained for evaluating up to 30 context policies at once (about 2 sec.).

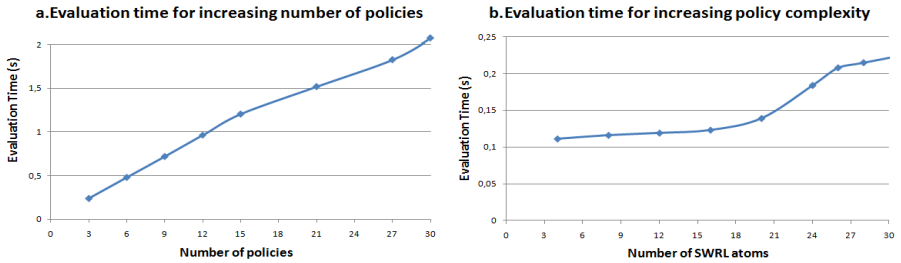

Fig 7. The EACM model GPIs/KPIs context policies evaluation results.

To determine the time needed for the *generation of the adaptation action sequence* when some GPIs/KPIs are broken, the EACM model was used to manage randomly generated service center context situations by means of reinforcement learning for about 27 hours (see Fig. 8). In the first 1000 decaseconds (das), almost all running times of the adaptation action selection algorithm are greater than 10 seconds. After that, the reinforcement learning mechanism begins to learn and achieving as a result the performance of having only four running times (the spikes in Fig. 8) greater than 10 seconds in the [5000, 7000] das time interval. Also, an overall reduction in the height of the peaks is visible because at each step the algorithm tries to determine the equivalent service center context situations and if equivalence is found the same sequence of actions is taken for execution.

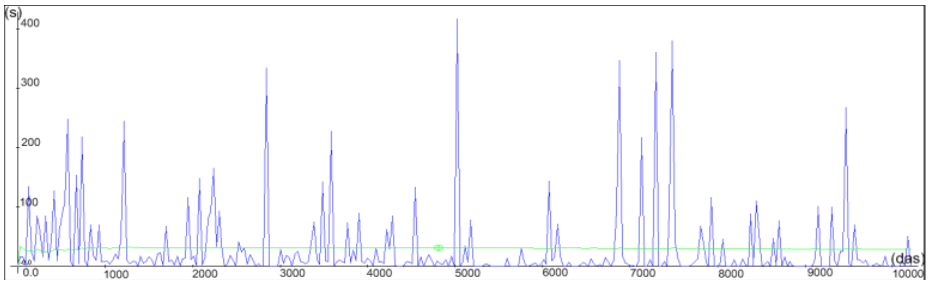

Fig 8. The EACM model adaptation action sequence generation.

To **evaluate the ontology representation of the EACM model,** three criteria were used: (i) instances creation time, (ii) instances retrieval time and (iii) memory usage. Our goal is to determine if the EACM model ontology implementation is feasible/scalable taking into account the above presented criteria and to identify the most suitable tool/API for the EACM ontology management. For ontology management, two strategies were used: (1) the ontology was created and persisted in the RAM memory (using Protégé and Kaon2 Ontology tools) and (2) the ontology was mapped onto a database model persisted in RAM memory. For the second strategy, three database models have been tested: HSQL relational model, Prevayler hierarchical model and Kaon 2 Database relational model.

To estimate the number of EACM model concepts instances that are created for representing different service centers, the following classification was used [17]: (i) small service centers having a number of servers between 101 and 500, (ii) medium service centers having between 501 and 5000 servers and (iii) large service centers with a number of servers over 5000. We consider that an average service center server has four processors each with two cores, four memory slots each of 2Gb and an array of four hard disks each with a capacity of 1Tb. For this type of server the EACM model instance ontology has to store approximately 17 concept instances: one IT Computing Complex Resource (corresponding to the server itself) and 16 IT Computing Simple Resource (corresponding to the server components). The number of instances corresponding to the service center IT infrastructure and the business tasks description are relatively low compared to IT Computing Resource instances,

hence they can be ignored for the scalability tests. We have defined an EACM Model management scenario in which a number of instances that have been automatically generated need to be administrated. The EACM model instance generation module takes as input an integer number n and automatically generates n^3 EACM instances for different ontology concepts. By choosing different values for n we have created a number of EACM instances which correspond to each service center type as follows: for n = 15 a number of 3,375 instances are generated corresponding to small service centers; for n = 25 a number of 15,625 instances are generated, corresponding to medium service centers, while for n = 50 and n = 70 result 125,000 and respectively 343,000 instances corresponding to large service centers.

The results presented in Fig. 9 show that: (i) the instance creation time and the memory usage grow exponentially for a number of instances higher than 50^3 but it remains within acceptable boundaries even for 70^3 instances and (ii) the instances retrieval time can be neglected for Prevayler, Kaon2 Ontology and Database.

Metric	Instance Creation Time (min:sec)				Time to retrieve all instances (min:sec)				Memory Usage (MB)			
Instance Count	15^3	25^3	50^3	70^3	15^3	25^3	50^3	70^3	15^3	25^3	50^3	70^3
Protégé	0:05	0:08	0:41	1:15	≈0:00	≈0:00	0:40	2:20	37	148	500	2577
Kaon2 Ontology	0:02	0:06	0:39	1:01	≈0:00	≈0:00	≈0:00	≈0:00	120	211	577	2750
HSQL + Hibernate	0:01	0:03	0:25	01:08	≈0:00	0:04	01:31	7:34	45	56	156	361
Prevayler	0:00	0:01	0:08	0:18	≈0:00	≈0:00	≈0:00	≈0:00	65	123	214	364
Kaon2 Database	0:03	0:07	0:22	01:00	≈0:00	≈0:00	≈0:00	≈0:00	70	110	409	634

Fig. 9. EACM ontology implementation management evaluation results.

5 Conclusions

This paper introduces the EACM model for representing the service centre energy related data in a uniform and machine interpretable manner. Reasoning based processes are defined and used on the model ontology representation to ensure energy awareness. The EACM model evaluation results are promising showing that the energy awareness capabilities can be enacted at a service center level in reasonable time frames using: (i) the model entropy to determine the service center greenness level and (ii) reinforcement learning to determine the adaptation actions to be executed when the defined greenness levels are not reached.

Acknowledgments

This work has been supported by the European Commission within the GAMES project [2] funded under EU FP7.

References

1. Kaplan, J.M., Forrest, W., Kindler, N.: Revoluzioning Data Center Energy Efficiency, McKinsey&Company, Technical Report (2008)
2. GAMES Research Project, http://www.green-datacenters.eu/
3. Wang, K.: Context awareness and adaptation in mobile learning, 2nd IEEE Int. Work. on Wireless and Mobile Tech. in Education, pp. 154-158, ISBN:0-7695-1989-X (2004)
4. Yu, Z.: iMuseum: A scalable context-aware intelligent museum system, Computer Communications, Vol. 31/18, pp. 4376-4382 (2008)
5. Burghardt, C., Reisse, C.: Implementing scenarios in a Smart Learning Environment, 6th Annual IEEE Int. Conf. on Perv. Comp. and Comm., ISBN: 0-7695-3113-X, (2008)
6. Pareschi, L., Riboni, D.: Composition and Generalization of Context Datafor Privacy Preservation, 6th Annual IEEE Int. Conf. on Perv. Comp. and Comm. (2008)
7. Grossniklauss, M.: Context Aware Data Management, 1st ed, VDM Verlag, ISBN 978-3-8364-2938-2 (2007)
8. Anderson, K. M., Hansen, F. A., Bouvin, N.: Templates and queries in contextual hypermedia, In: Proc. of the 17th Conf. on Hypertext and hypermedia, pp. 99 – 110 (2006)
9. Raz, D., Juhola, A. T.: Fast and Efficient Context-Aware Services, Wiley Series on Comm. Networking & Distributed Systems, ISBN-13: 978-0470016688, pp. 5-25 (2006)
10. Hofer, T., Schwinger, W.: Context-awareness on mobile devices – the hydrogen approach, the 36th Annual Hawaii International Conference on System Sciences, USA, pp. 292 (2003)
11. Cafezeiro, I., Hermann, E.: Ontology and Context, 6th Annual IEEE Int. Conf. on Perv. Comp. and Comm., ISBN: 0-7695-3113-X (2008)
12. Huaifeng, Q.: Integrating Context Aware with Sensornet, In: Proc. of 1st Int Conf. on Semantics, Knowledge, Grid, ISBN:0-7695-2534-2 (2006)
13. Sirin, E., Parsia, B.: Pellet: A practical OWL-DL reasoner, Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 5 (2), pp. 51-53 (2007)
14. Amoui, M., Salehie, M.: Adaptive Action Selection in Autonomic Software Using Reinforcement Learning, ICAC 2008, pp. 175-181, ISBN 0-7695-3093-1 (2008)
15. Bernstein, A., Kaufmann, E.: Querying the Semantic Web with Ginseng:A Guided Input Natural Language Search Engine, 15th Work. on Information Tech. and Syst. (2005)
16. Cioara, T., Anghel, I., Salomie, I.: A Generic Context Model Enhanced with Self-configuring Features, JDIM, Volume 7(3), pp.159-165, ISSN 0972-7272 (2009)
17 Material Stock in German Data Centres (2010) available at http://www.uba-green-it.de
18. Cioara, T., Anghel, I., Salomie, I.: A Policy-based Context Aware Self-Management Model, SYNASC 2009, pp. 333- 341, ISBN: 978-0-7695-3964-5 (2009).
19. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2(4), pp. 263-277 (2007).
20. Truong, H.-L., Dustdar, S.: A Survey on Context-aware Web Service Systems, International Journal of Web Information Systems, 5(1), pp. 5 - 31, Emerald, (2009).
21. Cioara, T., Anghel, I., Salomie, I., Pernici B.: A Context Aware Self-Adapting Algorithm for Managing the Energy Efficiency of IT Service Centers, UbiCC 2010, (2010).