Business Process Co-Design for Energy-Aware Adaptation

Cinzia Cappiello, MariaGrazia Fugini, Alexandre Mello Ferreira, Pierluigi Plebani, Monica Vitali
Politecnico di Milano
Via Ponzio 34/5 - 20133 Milano, Italy
{cappiell,fugini,ferreira,plebani,vitali}@elet.polimi.it

***Abstract*—Green IT mainly focuses on techniques to extend the products longevity or to virtualise physical resources as well as the provision of energy efficient hardware infrastructures. Less attention has been paid on the applications that run on the machines and their impact on energy consumption. This paper proposes an approach for enabling an efficient use of energy driven by the design of *energy-aware business processes*. *Energy-awareness* is given by an enrichment of a typical Business Process conceptual model with annotations able to support the assessment of the energy consumption of the involved business tasks. This information is the basis for the *energy-aware adaptation* to enact specific strategies to adapt process execution in case energy consumption needs to be lowered or energy leakages have been identified.**

***Index Terms*—Adaptive and context-aware processes, Service-oriented architectures for BPM, Resource management in business process execution, Green IT and energy-aware applications**

I. Introduction

Although sustainability has been recognized as an urgent problem, the IT community has been slow in acknowledging and tackling the problem from the applications viewpoint [1]. In fact, the research in this field has been mainly focused on techniques to extend the products longevity or to consolidate resources [2]. On the contrary, less attention has been paid to the application layer, where there is a potential for contributing to energy efficiency. For instance, to reduce the global energy consumption, an application could require less resources for its execution or adapt its behavior.

The goal of this paper is to propose an approach for designing *Energy-Aware Business Process* (E-BP) extending the typical Business Process (BP) conceptual model to capture the energy consumption of the involved business tasks. Energy consumption is constantly monitored by using specific indicators, called *Green Performance Indicators (GPIs)*, that have to be satisfied together with the more traditional functional and non-functional (i.e., QoS) requirements. Through *energy-aware adaptation* the E-BP is able to enact specific *strategies* to adapt its execution or structure in case energy consumption needs to be lowered or energy inefficiencies are identified.

Generally speaking, the innovative approach presented in this paper aims at providing a framework to design applications able to react in case the energy efficiency goals are not fulfilled. This framework is part of the GAMES European project. Figure 1 introduces the components and information

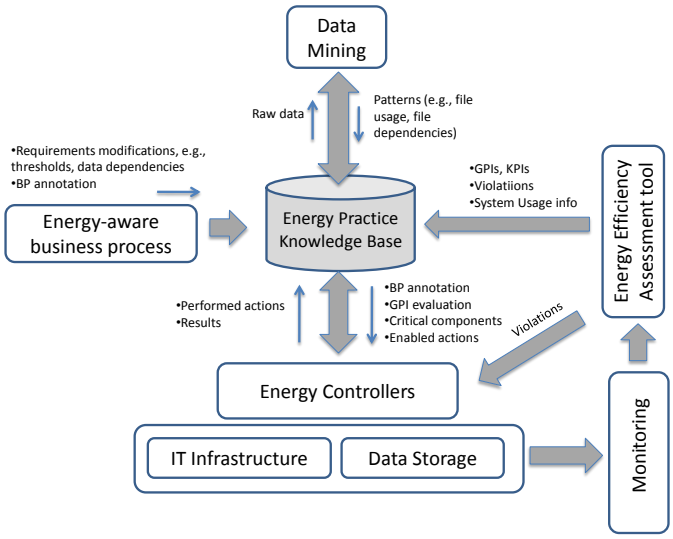

Figure 1. Information exchange in the GAMES architecture

that enable energy-aware adaptation in the GAMES architecture. Here, the adaptation is supported by information stored in the *Energy Practice Knowledge Base*. This knowledge base is initially fed with data about the BP that are executed in the service data center. These data, defined at design time and exploited at run-time, include: requirements on GPIs, requirements on traditional quality dimensions, and the configuration of the infrastructure on which the BP runs. The *monitoring system* is configured to monitor the service center components (e.g., IT infrastructure, data storage) and to transmit the sensed information to an *assessment tool* that is in charge to evaluate GPIs/QoS dimensions, raise exception in case of GPIs/QoS non-fulfillment, and consequently alert energy controllers in case critical violations occur. The knowledge base collects also this information and maintains a historical log. Based on this knowledge, the *Energy controllers* are designed to properly react to recover problematic situations. They enact suitable adaptation strategies on the basis of the characteristics of the available strategies and information gathered from the knowledge base such as results of past adaptation actions, properties of the running applications, relevant patterns retrieved by data mining applications (e.g., correlations or dependencies). The selection of the most suitable strategy should guarantee energy savings and the satisfaction of QoS requirements.

The paper is structured as follows. Firstly, Section II depicts

TABLE I
GPI AND QOS REQUIREMENTS

ID	Name	Description	Formula
GPI1	Energy Aware Application Performance	Measures the number of transactions executed within one kWh	$\frac{numberOfTransactions}{kWh}$
GPI2	CPU usage	Measures the CPU load	$\frac{amountOfCPUused}{amountOfCPUallocated}$
GPI3	Storage Usage	Measures the storage load	$\frac{diskSpaceUsed}{totalDiskSpaceAllocated}$
GPI4	IOPS/Watt	Number of I/O operations per second per watt	$\frac{numberOfI/Ooperations}{Watts}$
GPI5	Memory usage	Measures the memory load	$\frac{amountOfMemoryUsed}{amountOfMemoryAllocated}$
QoS1	Response time	Time to react to a given input	$t_{response} - t_{request}$

a working example that will be used throughout the paper to describe the presented approach. Section III presents the co-design approach that is based on the relationships between the E-BP and the devices that consume power. These relationships allow the designer to estimate the energy footprint of an E-BP. Section IV discusses how the estimated consumption is compared to the actual one and Section V introduces the adaptation strategies that can be enacted in case energy savings are required. Examples that show how "green" process requirements are able to trigger adaptation strategies at different infrastructure layers are illustrated in Section VI.

II. SCENARIO

To motivate and better illustrate the presented approach we present a possible application scenario representing a simple and generic BP for on line book sales shown in Figure 2.

A generic user connected to the bookstore browses its contents and selects some items. If s/he desires to buy the selected items s/he inserts personal information in the system. If the user is already registered the customer information is retrieved from the database, otherwise a new entry in the database is created. Information about the order is reviewed before proceeding with the payment. At this stage the customer can confirm the order, pay and proceed toward delivery, or s/he can cancel it. Each task in the BP corresponds to one or more software services which run on a virtual environment related to some physical devices.

As discussed before, the BP execution has to satisfy both QoS and energy consumption requirements. The set of GPIs and QoSs and relative constraints are defined by the designer and depends on the application. GPIs and QoS requirements defined for the described scenario are listed in Table I and correspond to the indicators chosen for monitoring applications. The chosen indicators may have a direct or indirect impact on energy consumption (most of them are not directly related to energy in their formula). For instance, usage indicators are not directly related to energy consumption but they can help to reach a good resource allocation that results in energy saving.

III. CO-DESIGN OF ENERGY-AWARE BUSINESS PROCESSES

An E-BP is a service-based BP in which the activities are defined not only in terms of their functional and non-functional requirements, but also with specific annotations able to provide useful information for guiding the energy assessment and the selection of more suitable adaptation strategies. These annotations are metadata describing the properties of the whole process and the composing tasks: process relevant application data, temporal constraints and resources requirements. More in detail, we consider the following metadata:

- Flow metadata: provide information regarding the BP control flow and thus the execution of certain activities in a process;
- Energy and performance constraints: refer to energy and performance conditions or constraints within process flows;
- Resource metadata: provide information regarding the used resources when executing a certain task;
- Data metadata: provide information regarding the data used throughout a process.

Considering the on-line books sales process example introduced in Section II, the application will be annotated as shown in Figure 2. Here, the set of services able to perform the activities are deployed on three different virtual machines (i.e, VM1, VM2, VM3) that are installed on two physical servers (i.e., server1, server2). Inputs and outputs of activities are annotated with the characteristics of data. The constraint about *GPI1* is associated with the process while constraints about *QOS1* are associated both with the whole process and single activities. Requirements about *GPI2*, *GPI3*, *GPI4* and *GPI5* are not reported in the figure since they are strictly related to the infrastructure layer.

Assuming that the designer is able to define the functional and non-functional aspects of a process, focusing on the energy aware aspects, his/her goal is to design an E-BP that minimizes the energy consumption without affecting the QoS constraints, exploiting the metadata coming from the data mining applications. Since the E-BP energy footprint depends on the virtual environments involved, their configuration, and how the applications are deployed on them, then the definition of a relationship between the deployment configuration and the energy consumption is required. The result of the co-design phase is an E-BP annotated with all the information about the execution, the deployment, and the energy footprint.

To compute the E-BP energy footprint, we consider a system model composed of three different layers: *infrastructure layer*, *middleware layer*, and *application layer* (Figure 3). Generally speaking, at the infrastructure layer it is possible to measure the real power consumption of physical devices $\{pd_i\}$, whereas at the middleware and application layer, the energy consumption attributed to virtual environments $\{ve_j\}$ and service applications $\{s_k\}$ has to be inferred from the measured data.

More in details, the infrastructure layer includes all the physical energy-hungry equipments installed in the IT service

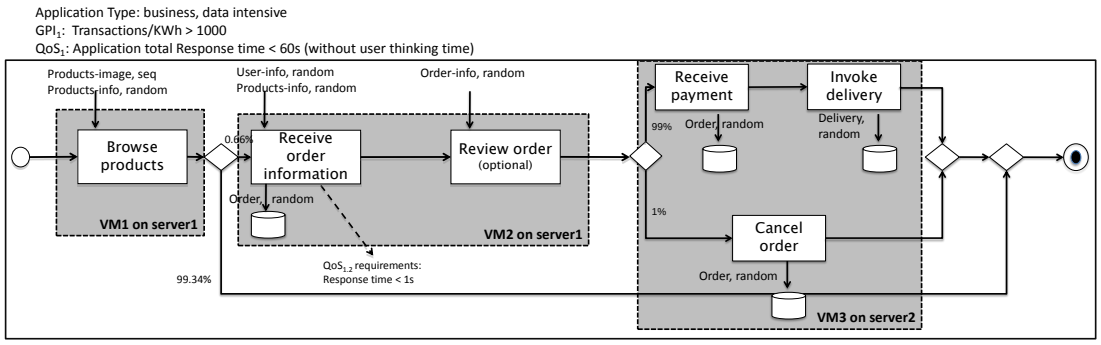

Figure 2. Annotated on-line Book sales process.

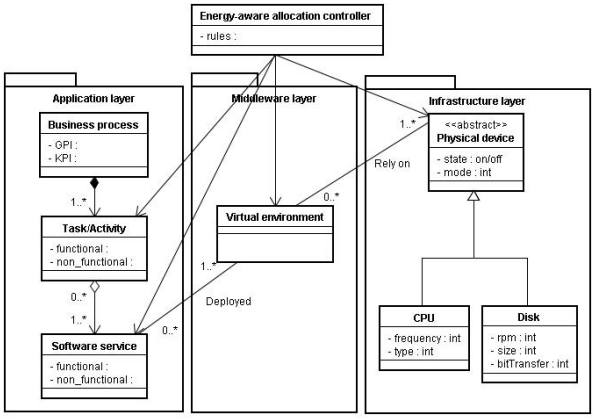

Figure 3. System model.

center. In particular, in our work, we focus on the power consumed by the server machines that can be measured directly on the devices. Currently, there are also some software-based solutions that provide this information. For instance, tools as PowerInformer[1] or JouleMeter[2] provide information about the power consumed by the components of a server as, for instance, the CPUs and the disks.

The middleware layer is responsible to manage virtual resource reservation, workload distribution, and resource monitoring. As shown in Figure 3, the virtual environment is composed of a set of limited computational resources associated with a physical device. Some work in the literature [3], [4] advocate the possibility to infer, even instantaneously, the power consumed by a virtual environment by considering the resources usage. This allows the designer to estimate, given a virtual environment ve_j, the maximum power consumed at run-time. This estimation can be done empirically, by full loading the execution of the virtual environment and thus assessing the peak power consumption of the VM. We can state that the power consumed by the virtual environment P_{ve_j} depends on the configuration of such a virtual environment C_{ve_j} and power of the physical devices $\{P_{pd_i}\}$ on which the virtual environment is executed:

$$P_{ve_j}(t) = f(C_{ve_j}(t), \{P_{pd_i}(t)\}) \quad (1)$$

Note that P_{pd_i} is a value gathered from the continous monitoring of the system.

Finally, the application layer includes the BPs that embrace software services. In our model, a BP is composed of a set of tasks/activities performing the functions and also described by their non-functional requirements. As already stated in Section I, non-functional requirements include constraints on QoS dimensions (e.g., maximum response time) and GPIs (e.g., maximum energy consumption). Given one task, in our assumption, one or more software services are available to perform it, also satisfying its non-functional constraints.

In this model, for the sake of simplicity, we assume hereafter to have only one E-BP running, with several instances, on our IT service center. In our future work, we will deal with a more complete scenario where multiple E-BPs concurrently run on the same IT service center with shared resources.

About power metering, the designer needs to know which is the power consumed by an application running on a virtual machine. This application is in charge of executing one or more activities composing the E-BP. Similarly to what we did at the middleware layer, we start from the assumption that software tools exist (e.g. [5] [3]) able to measure the power consumed by a single application. The designer can preliminarily run the service to evaluate which will be the maximum power $P_{s_k}(t)$ it requires during the execution. As occurred for the virtual machine power that depends on the physical machine power, in this case, the power required by an application providing a service will depend on the virtual machine where the application runs. Thus:

$$P_{s_k}(t) = g(P_{ve_j}(t)) \quad (2)$$

The three layers are also considered in the definition of GPI/QoS indicators. In our approach, we propose to classify GPIs and QoS requirements as *high level indicators* associated with the application and middleware layers, and *low level indicators* related to the Infrastructure layer. An indicator can be defined as aggregation of (or can be influenced by) different elementary variables or indicators defined at a lower level.

[1]http://software.intel.com/en-us/articles/intel-powerinformer/

[2]http://research.microsoft.com/en-us/projects/joulemeter/default.aspx

[3]see PowerTop (http://www.lesswatts.org/projects/powertop/)

The relationship between the deployment configuration and the power consumption helps the designer to realize which is the maximum energy consumption of the E-BP. To this aim, the designer needs to know for each task: (i) the service s_k that performs the task, and (ii) the execution time $\overline{t}_{s_k}$. The total energy consumption estimated for an E-BP, i.e., $\overline{E}_{BP}(T)$, depends on how services are organized. Eq. 3 shows how to calculate the energy consumption of services based on the process flow. Given a set of services $S = \{s_k\}$, if they are executed in sequence or in parallel, the energy results from the sum of the energy consumed by the tasks (see row *I*). In case the services are in different mutual exclusive branches, the energy depends on the estimation of the probability to execute the branches (see row *II*). Finally, in case of iterations, the energy consumed depends on the number of iterations k (see row *III*) [20]. This number can be obtained by applying unfolding techniques on previous executions analysis [6]. Of course the obtained power consumption values refer only to the BP based on its selected services and do not represent the consumption of the physical device.

$$\overline{E}_S(T) = \begin{cases} \textit{I (sequence)} : \sum_{s_k \in S} \int_{t_{s_k}} P_{s_k}(t) \cdot dt \\ \textit{II (alternative)} : \sum_{s_k \in S} p_{s_k} \cdot \int_{t_{s_k}} P_{s_k}(t) \cdot dt, \\ \quad \text{where } p_{s_k}\text{: probability of execution for} s_k \\ \textit{III (iteration)} : k \cdot \sum_{s_k \in S} \int_{t_{s_k}} P_{s_k}(t) \cdot dt \\ \quad \text{where } k\text{: number of iterations} \end{cases} \tag{3}$$

Considering the scenario described in Section II, the BP is composed of six activities that are executed by means of six different services $[s_1 \dots s_6]$. Considering the resources reservations, the process flow and the annotations specified in Figure 2, the average energy estimated for the process can be defined as the following:

$$\overline{E}_{BP}(T) = \int_{\overline{t}_{s_1}} P_{s_1}(t) \cdot dt + 0.0066 \cdot \Big(\int_{\overline{t}_{s_2}} P_{s_2}(t) \cdot dt + \int_{\overline{t}_{s_3}} P_{s_3}(t) \cdot dt + 0.99 \cdot \Big(\int_{\overline{t}_{s_4}} P_{s_4}(t) \cdot dt + \int_{\overline{t}_{s_5}} P_{s_5}(t) \cdot dt \Big) + 0.01 \cdot \int_{\overline{t}_{s_6}} P_{s_6}(t) \cdot dt \Big) \tag{4}$$

A deeper discussion on energy estimation is out of the focus of this paper. For further information about estimation techniques it is possible to refer to [7][8][9].

IV. Run-time energy assessment

The real power consumption of an application can be obtained by processing data gathered from sensors installed on the physical devices. According to the power consumed by each device and the resources assigned to the components at the different layers and their usage, we can obtain the real energy consumption $E_{BP}(T)$ and the evaluation of the other selected GPIs/QoS dimensions by mining the data saved in a log file. An example of real power consumption of a process is represented by the dashed line in Figure 4.

The availability of data about the power estimated and consumed at run-time allows the designer to identify energy inefficiencies i.e., energy leakage. *Energy leakage* allows identifying the resources that are not working in the best possible way and it is defined as the difference between the actual energy consumption $E_{BP}(T)$ related to the process

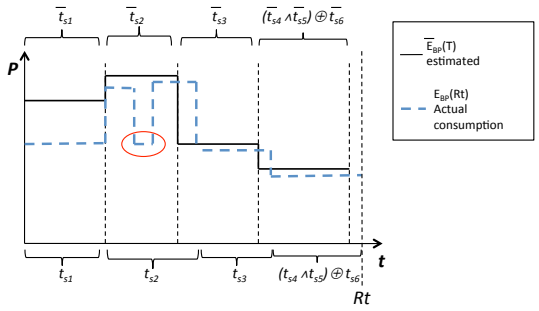

Figure 4. Power estimation and consumption (Area represents the energy).

execution and the estimated one $\overline{E}_{BP}(T)$. For example, the case in which $E_{BP}(T) < \overline{E}_{BP}(T)$ might (i) be a sign that a source is not working properly and thus wasting energy or (ii) reveal an error in the resource reservation process. In the former case, the source of this leakage has to be identified between resources allocated to the process. The latter case occurs when, for example, during process execution all the instances consume less energy than expected but GPIs/QoS parameters are satisfied. In this situation it is possible to state that the amount of reserved resources is overestimated.

In the situation represented in Figure 4 an energy leakage occurs. For some tasks $E_{BP}(T) < \overline{E}_{BP}(T)$ and response time associated with the process is greater than expected. The most difficult issue is the definition of (i) the task responsible of the leakage and/or (ii) the inefficient resource (e.g., processor or memory inefficiency). The identification of the responsible task is not trivial since the duration of the tasks might be shorter or longer than expected and thus the active task in a specific time instant could be different from the task expected in the execution plan. In fact, changes in the process execution can affect the estimated execution time of the different tasks. Let us suppose that during the execution of Task 2, the disk controller decides, for internal reason, to slow down the disk access from normal to quiet mode (see the circle in Figure 4). This can reduce the energy consumption but may increase the actual response time (Rt) of the whole application. So, as the calculation of the actual energy should be based on the actual execution time, than a lower functioning mode of the storage is not always associated with a lower energy consumption since it decreases the power but may increase the execution time. Indeed, if we just compare at a given time if the estimated power is lower than the actual one, then Task 3 should be blamed of the leakage instead of Task 2. To solve the situation, both curves have to refer to the same time line by tightening or relaxing one of them. Anyway, in this situation it is possible to say that Task 2 is consuming more than expected and thus it could be the source of the problem. It is also possible to notice that Task 1 is consuming less than expected revealing an overestimation of the power consumption. In fact, notwithstanding this divergence with the expected energy consumption, the response time of the service is still satisfied. In both cases, adaptation strategies can improve the energy efficiencies of the service center for these applications (Section V).

V. ENERGY-AWARE ADAPTATION

An E-BP is *adaptive with respect to energy consumption* when the amount of resources needed can be adapted so that the E-BP can run maximizing the use of the resources and minimizing the power consumption. Such adaptivity regards the flexibility in the management of the resources available.

As shown in Section IV, adaptation can be triggered by the violation of a GPI/QoS requirement (*reactive* adaptation). However, if the process consumes less than expected, resources that are uselessly allocated should be set free (*proactive* adaptation) by, for instance, reconfiguring the VMs. Both energy-aware adaptations types can be implemented by using a set of available strategies classified on the basis of their impact on the system as: (i) *Less quality* strategies that rely on non-functional requirements reduction; (ii) *Less functionality* strategies that regard to computational or data information reduction; (iii) *Resource reallocation* strategies that change how/which resources are used by the process. Some strategies may combine all the three classification together.

Table II sums up some of the energy-aware adaptation strategies that we consider in our approach. Strategies may be applied at design-time (e.g., process re-design) or run-time (e.g., enable energy aware mode). When adaptation is required, strategy selection is based on the strategy characteristics and historical information gathered from log files. In particular, as described in Section I, our decisions are based on the Energy Practice Knowledge Base containing the results of the past adaptation actions together with the description of the critical situation that they were expected to solve.

If adaptation is triggered by GPI/QoS violation, it might happen that more than one adaptation strategy could be enacted. In our approach, the selection of the more suitable strategy is based on the definition of a set of adaptation rules $\mathcal{R}$ that link the violation of an indicator with the set of associated adaptation strategies. More formally, each indicator $I_{h,obj}$ (a GPI or a QoS) associated with a system object $obj \in \{\text{E-BP}, \{s_k\}, \{ve_j\}, \{pd_i\}\}$ can be defined in terms of a name which uniquely identifies the indicator, and a formula which contains the specification of the evaluation algorithm. An adaptation rule $R_{h,obj}$ can be defined as:

$$R_{h,obj} = \langle I_{h,obj}.name, \{\langle C_{hw,obj}, \langle As_{hwm}, Conf_{hwm}, Imp_{hwm}\rangle\rangle\}\rangle \quad (5)$$

where: $C_{hw,obj}$ is a set of constraints delimiting the admissible values of $I_{h,obj}$ for the specific system object obj; As_{hwm} is the set of adaptation strategies to be enacted in case of violation of the corresponding constraint $C_{hw,obj}$; $Conf_{hwm}$ is the confidence associated with the effective execution of the action associated to the violation; and Imp_{hwm} is the degree of the importance of the action that depends on the impact that it has on the energy consumption state of the system.

We assume that strategies enactment is based on the possibility to distinguish between high level indicators and low level indicators as defined in Section III and on the identification of relationships among indicators. In fact, it might happen that an indicator $I_{h,obj}$ is influenced by other indicators. Thus, it is possible to identify a function $Dep : I \times \mathcal{P}(I) \rightarrow \mathcal{P}(I)$ that for each indicator $I_{h,obj}$ identifies the set of correlated indicators $I'_{l,obj} \subset I$. Dependencies between indicators imply that a modification in the value of an indicator results in a modification in the values of the dependent indicators. Relations among indicators can be computed using data mining techniques that are not discussed here.

TABLE II
ENERGY-AWARE ADAPTATION STRATEGIES LIST

Energy-aware strategies	Description	Layer	Type
Process re-design	Re-definition of the process functionalities	Application	Reallocation
Process structure change	Re-definition of the process workflow	Application	Reallocation
Enable Energy Aware Mode (EA Mode)	Processes run in energy aware mode so that optional task can be skipped or task functionalities can be limited (e.g., less operations or less data)	Application	Less quality Less functional
Enable service switching off	if the same task is offered by two different services, one of them can be switched off in order to save energy	Application	Less quality
SLA re-negotiation	Re-negotiate to reduce functional and non-functional minimum requirements	Application Middleware	Less quality Less functional
Service container substitution	Redo the matchmaking in order to find out a more energy-friendly service	Middleware	Less quality
Container migration	Migrate or change the process instances to another application container	Middleware	Reallocation
Switching mode	Change of the processor or storage mode	Middleware	Reallocation
VM reconfiguration	The VM associated with the service is reconfigured	Middleware	Reallocation
VM replication	activation of another VM which hosts the same service	Middleware	Reallocation
Enable switching off	Enable the possibility to switch off servers or disks	Infrastructure	Reallocation
Enable data migration	Enable the possibility to migrate data from an array disk to another one chosen taking in considerations similarities between data access rate or data coupling	Infrastructure	Reallocation

Algorithm 1 details an iteration of the adaptation strategy selection and enactment. At run time, the monitoring module gathers all the data necessary to compute the values of all the indicators $Val_{h,obj}$ and to evaluate the associated requirements. The evaluation of requirements can be seen as a function $Eval : R \times Val \rightarrow \{AS, \emptyset\}$, which, given a value for $I_{h,obj}$, checks each constraint $C_{hw,obj}$ (line 29) and, in case of violation, returns the strategy As_{hwm} associated with the highest importance and confidence values (line 35). If more than one indicator is violated, the system starts considering the ones at the higher level one by one (line 2). In order to avoid failures or performance reduction at the application level, relationships among indicators are exploited (line 12). In fact, the function $Eval$ will be applied to all correlated low level indicators $I'_{l,obj}$: $Eval(R_{l,obj}, val_{l,obj})$ (line 15). Once all the adaptation strategies have been applied, the value of $I_{h,obj}$ should satisfy all constraints, that is, a second

evaluation of $Eval$ should not enact any other action for the considered indicator (line 17). If instead $Eval$ tries to enact other strategies this means that adaptation actions did not succeed. In this case, the function $Eval$ will be applied directly to the examined indicator (line 23). If despite the activation of all the adaptation actions, the value of $I_{h,obj}$ still violates constraints, a human intervention is required and the application should be redesigned or restructured (line 10). Some adaptation strategies could negatively affect system performances (e.g., EA mode, switching mode). In this cases the adaptation strategies can be executed only for a predefined time interval or until when the analyzed indicator satisfies all the related constraints.

Algorithm 1 Adaptation Strategy Enactment Algorithm

Require: $rule_{*,obj}$ (all the rules related to object obj)
Ensure: adaptation strategies enactment to eliminate indicators' violation
1: **function** PROCEDURE_ENACTMENT($rule_{*,obj}$) {
{the procedure is continuously executed during the assessment phase}
2: sort $rule_{*,obj}$ from High to Low level based on indicators
3: **for all** $rule_{h,obj} \in rule_{*,obj}$ **do**
4: $ind_h \leftarrow$ related indicator instance from $rule_{h,obj}$
5: $value_{h,obj} \leftarrow Val(ind_h, obj)$ //get indicator value from assessment tool
6: $eval_h \leftarrow$ EVAL($rule_{h,obj}$, $value_{h,obj}$)
7: **if** $eval_h = \emptyset$ **then**
8: do nothing //there is no violation in $rule_{h,obj}$
9: **else if** $eval_h$ = human intervention required **then**
10: message(human intervention required for $rule_{h,obj}$)
11: **else**
12: **for all** $indLow_l \in$ Dep(ind_h) **do**
13: $rule_{l,obj} \leftarrow$ associated rule with $indLow_l$
14: $value_{l,obj} \leftarrow Val(indLow_{l,obj})$
15: enact adaptation strategy from EVAL($rule_{l,obj}$, $value_{l,obj}$)
16: update $value_{h,obj}$
17: **if** EVAL($rule_{h,obj}, value_{h,obj}$) = $\emptyset$ **then**
18: exit FOR cycle
19: **end if**
20: **end for**
21: **if** EVAL($rule_{h,obj}, value_{h,obj}$) $\doteq$ adaptation strategy **then**
22: //apply adaptation directly to the examined indicator
23: enact adaptation strategy from $eval_h$
24: **end if**
25: **end if**
26: **end for**
27: }
28: **function** EVAL(rule, value): STRATEGY {
29: check that indicator's value satisfies constraints
30: **if** there is one or more violated constraints
31: select the best adaptation strategy not used yet
32: **if** there is no remained strategy
33: **return** human intervention required
34: **else**
35: **return** adaptation strategy
36: **end if**
37: **end if**
38: **else return** $\emptyset$
39: }

VI. Case study discussion

In this section we show an application of the presented approach starting from the BP introduced in Section II. The first task/service (Browse Products), the most relevant of the whole process due to the branching probabilities, is taken into account in our first attempt to describe possible scenarios for energy saving improvement. The set of rules defined for the described case study is summarized in Table III.

GPIs can be organized in an oriented graph representing relations between them. In the example, GPI2 is related to GPI1, because a reduction of the CPU usage implies an increasing of the Application Performance. Instead of reducing the quality of service enabling the EA mode or switching off a server, the same result could be obtained reducing the CPU usage by switching off some slightly used processors applying Rule 2.1 instead of Rule 1.1 and Rule 1.2. We could also optimize storage usage and allocation to reduce energy consumption. In this case GPI3 and GPI4 are involved, so rules Rule 3.1, Rule 3.2 and Rule 4.1 can be applied. The choice of the best rule in a given context depends on past observations and on confidence and impact values for each rule.

TABLE III
Case study rule definition

Rule ID	GPI/QoS	Constraint	Rule Action and Description
Rule1.1	GPI1	>10000	*enable Energy Aware Mode (EA Mode)*: this action has effects on the Browse Product task and the result will probably be a reduction of the shown results in the research for a product (e.g. ten results instead of twenty)
Rule1.2	GPI1	>10000	*enable service switching off*
Rule2.1	GPI2	>80%	*enable the switch off of all the other servers in the rack*: this rule is used to reduce the CPU and memory usage on a given machine
Rule2.2	GPI2	>80%	*VM reconfiguration*: e.g. decreasing the amount of CPU assigned to the service instance
Rule3.1	GPI3	>30%	*enable data migration*
Rule3.2	GPI3	>30%	*enable empty array disks switch off*
Rule4.1	GPI4	>100	*enable acoustic mode*: disks can be configured to reduce the acceleration and velocity of the disk head
Rule5.1	GPI5	>75%	*VM reconfiguration*: e.g. decreasing the amount of memory assigned to the service instance
RuleQoS1.1	QoS1	<1 sec	*enable an alternative service*: this rule allows to have more than a service for the same task in order to reduce response time
RuleQoS1.2	QoS1	<1 sec	*VM reconfiguration*, e.g. increasing the amount of memory dedicated to the service
RuleQoS1.3	QoS1	<1 sec	*VM replication*: this VM could be on a different server so it could cause the switching on of a server

In order to prove the effectiveness of the approach, the case study has been executed and monitored to obtain data about resources usage and application performance. The BP described in Figure 2 has been simulated by an open source implementation of the TPC-C benchmark: the TPC-C Uva [10]. In fact, it is easy to find a correspondence between the described activities and the TPC-C ones. An instance of the application has been installed and executed on a Linux VM running over a Windows XP host through VMWare hypervisor. The benchmark has been executed with several VM configurations. Data about resource utilization, needed to compute GPI2, GPI3, GPI5 and partially GPI4 are taken from the VM operating system during two hours tests with a five minutes sampling period. The number of transactions and response time for each task of the benchmark have been

TABLE IV
GPI AND QOS AVERAGE VALUES WITH DIFFERENT VM CONFIGURATIONS

TestId	GPI1 (App. Perf.)	GPI2 (CPU us.)	GPI3 (Storage us.)	GPI4 (IOPS/Watt)	GPI5 (Memory us.)	QoS1 (Resp. time)
Test1	13822	47.15%	38%	133	97.96%	0.034
Test2	15080	67.42%	32%	175.57	93.30%	0.063
Test3	16250	99.67%	38%	107.42	97.91%	0.048
Test4	17354	99.67%	38%	190.14	97.96%	0.054
Test5	14188	94.21%	38%	111.4	54.61%	0.078

monitored using information provided by the benchmark itself and can be used to compute QoS1 and part of GPI1. The remaining needed information related to energy consumption have been collected using JouleMeter (for I/O energy consumption needed for GPI4) and PowerTop (for application energy consumption needed for GPI1) tools. Data have been collected and then analyzed for different VM configurations: (*Test1*) 4 CPUs, 512 MB of memory, 10 GB hard disk; (*Test2*) 3 CPUs, 512 GB of memory, 10 GB hard disk; (*Test3*) 2 CPUs, 512 MB of memory, 10 GB hard disk; (*Test4*) 1 CPU, 512 GB of memory, 10 GB hard disk; (*Test5*) 2 CPUs, 1 GB of memory, 10 GB hard disk. Table IV reports average values for GPIs and QoS constraints for each test.

Starting from the *Test1* configuration we can explain how the adaptation works. The assessment tool checks if all the indicators satisfy the given constraints: GPI2 is not satisfied because CPU usage is out of the boundaries. The service is executed on a single VM, so the only rule that can be used to increase the value of GPI2 is Rule2.2 in Table III. The effect of rule application is the decreasing of CPUs allocated from four to three, obtaining the *Test2* configuration. Even in this configuration GPI2 is not satisfied, so another reconfiguration can be enacted reducing CPUs from three to two. At this point we obtain the *Test3* configuration and all the indicators are satisfied. To avoid service unavailability, if the service is offered by a single VM two steps are required: at first a new VM with the new configuration can be started and the service can be deployed over it; after that the old VM can be powered off. In the same way, starting from the *Test5* configuration, GPI5 is not satisfied and requires the application of Rule5.1 to reduce memory allocation from 1 GB to 512 MB. The resulting configuration is the one of *Test3* again.

From the results in Table IV it is easy to note the relation between GPI1 and GPI2: decreasing the number of CPUs allocated to a Virtual Machine, the Application Performance indicator increases because of a minor energy consumption of the machine. The same behavior can be observed with GPI1 and GPI5. Measures over VM in *Test1* and in *Test3* have shown that the average energy saving per hour between the two configurations is between 4-5 Watt-hours, while comparing *Test5* and *Test3* we gain between 3-4 Watt-hours per hour. To this gain we can add the one due to resources releasing that can be reused for other purposes, the application of the process to all the services in the data center and the cascade effect for which very few watts saved at the VM level may cause significant impact in the overall data center energy costs.

VII. RELATED WORK

Energy-related issues have been addressed by various researchers ranging from the architecture and power management communities to the data management and software systems groups [11]. However, energy-awareness still needs to find methods and tools in the area of BP management.

In [1], authors discussed how organizations have recognized environmental sustainability as an urgent problem but still the Information Systems (IS) academic community is slow in acknowledging and tackling the problem from the software and applications viewpoint. The paper proposes ways for the IS community to enter in the development of environmentally sustainable business practices. Considering a more practical approach, [12] discusses the organizational supply chain and proposes a sustainable IS management framework, which delineates some steps based on a resource identification view. A green IT framework is also proposed by [13] which provides clearly defined concepts related to green information technology and devises a practical implementation for sustainability-based feedback mechanisms. A resource model is proposed by [14] to minimize carbon footprint emissions from a BP management level. The authors' approach annotates CO2 emission for each single activity composing the BP in order to figure out resource costs within an "usage-cost relationship" model and to re-design the process with same functional goals while reducing carbon footprint.

One of the most common solutions to reduce power consumption at the middleware layer is the server consolidation technique. Although we are not providing any solution in the server virtualization field, i.e., at the middleware layer, our proposed approach is built on top of existing mechanisms that provide relevant information about the infrastructure layer, in particular, information about power consumption and resource allocation/usage. Such approaches tackle the problem at a lower level, like the heterogeneous resource identification proposed by [15]. The authors basically aim to use slow servers to perform non-critical tasks in order to reduce the service cost. In order to evaluate performance and power consumption, a new metric is proposed by [16] which quantify the difference between minimal resources required by a task and what the system actually allocates. Such metric tries to reduce the number of used servers and increase the use of the most energy efficient ones in order to reduce the gap between server capabilities and service requirements. Similar approaches comprising power-aware strategies at the middleware layer can also be found in [17], [18].

In our previous work [19], we have proposed novel energy-aware resource allocation mechanisms and policies for BP-based applications. These mechanisms were intended to minimizing the energy consumption of the process layer, the infrastructure layer, and the control layer of a data center. We have presented a new energy efficiency metric for a single service, which maps directly the relationship between

energy consumption and execution time [20]. By being able to compute both quality and energy metrics for each service, we have designed a service-based process by executing a novel constraint-based quality and energy-aware service composition algorithm.

VIII. Conclusion

In this paper, we have presented a model that enables evaluating energy consumption in the processes layer starting from the analysis of the characteristics of the activities composing the process and the resources in use. On the basis of the actual use of the resources during execution, a way to improve energy efficiency of the process is proposed. Our approach enables the identification of energy leakages and/or of GPIs/QoS violations and the selection of suitable adaptation actions that can be applied to the process and its virtual execution environment to maximize the use of the resources and minimize the power consumption.

The collected results demonstrate the effectiveness of the proposed framework. By analyzing the context relevant information and selecting suitable adaptation strategies we were able to improve energy efficiency of a simple application. In our example, with five different test bed configurations and one BP, we have shown the tight correlation among the indicators, highlighting the relevance of the adaptation strategy selection module. Initial results encourage us to further investigate the problem in the direction of energy-awareness process annotation.

First of all, the process annotation could consider a richer activity profile, for instance considering data used in the process, their relevance for the process, the type of access of data and data dependencies. This thorough process analysis could be useful to support process evolution by considering how changes in the structure of the process can make a process execution more efficient with respect to energy efficiency, still maintaining the original functionality and quality of service required for the process. Furthermore, as the approach now considers only a single BP running alone in the data center, future work will investigate how to manage a more realistic situation in which several processes are running concurrently. Finally, we will analyze the dependencies between physical devices, operating systems and Business Processes in order to identify the global effects of adaptation strategies (i.e., positive and negative impact on the system components) and improve their selection. Also, the overall energy impact of the introduced monitoring and energy controllers systems will be analyzed as well.

Acknowledgment

This work has been partially supported by the GAMES project (http://www.green-datacenters.eu/) and has been partly funded by the European Commission's IST activity of the 7th Framework Program under contract number ICT-248514. This work expresses the opinions of the authors and not necessarily those of the European Commission.

References

[1] R. Watson, M.-C. Boudreau, and A. Chen, "Information systems and environmentally sustainable development: Energy informatics and new directions for the is community," *MIS Quarterly*, vol. 34, no. 1, pp. 23–38, 2009.

[2] T. Velte, A. Velte, and R. Elsenpeter, *Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line*. McGraw-Hill, 2008.

[3] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proc. of SoCC*. New York, NY, USA: ACM, 2010, pp. 39–50. [Online]. Available: http://doi.acm.org/10.1145/1807128.1807136

[4] A. E. Husain Bohra and V. Chaudhary, "VMeter: Power modelling for virtualized clouds," in *Proc. of IPDPSW*. IEEE Computer Society, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1109/IPDPSW.2010.5470907

[5] T. Do, S. Rawshdeh, and W. Shi, "pTop: A process-level power profiling tool," in *Proc. of HotPower*. ACM, October 2009.

[6] L. Zeng, B. Benatallah, A. H.H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, 2004.

[7] B. Krishnan, H. Amur, A. Gavrilovska, and K. Schwan, "Vm power metering: feasibility and challenges," *SIGMETRICS Perform. Eval. Rev.*, vol. 38, pp. 56–60, January 2011. [Online]. Available: http://doi.acm.org/10.1145/1925019.1925031

[8] J. F. G. and T. Ledoux, "Self-optimisation of the energy footprint in service-oriented architectures," in *Proc. of the 1st Workshop on Green Computing*, ser. GCM '10. New York, NY, USA: ACM, 2010, pp. 4–9. [Online]. Available: http://doi.acm.org/10.1145/1925013.1925014

[9] R. Bertran, Y. Becerra, D. Carrera, V. Beltran, M. Gonzalez, X. Martorell, J. Torres, and E. Ayguade, "Accurate Energy Accounting for Shared Virtualized Environments using PMC-based Power Modeling Techniques ," in *The 11th IEEE/ACM International Conference on Grid Computing (GRID-2010) Month = October, Year = 2010*.

[10] D. Llanos and B. Palop, "TPCC-UVa: an open-source TPC-C implementation for parallel and distributed systems," in *Proc. of IPDPS*. IEEE, 2006, p. 389.

[11] S. Harizopoulos, M. Shah, and P. Ranganathan, "Energy efficiency: The new holy grail of data management systems research," in *Proc. of CIDR*, 2009.

[12] N.-H. Schmidt, K. Erek, L. M. Kolbe, and R. Zarnekow, "Towards a Procedural Model for Sustainable Information Systems Management," in *Proc. of HICSS*. Hawaii, USA: IEEE Computer Society, 2009, pp. 1–10.

[13] H. Mann, G. Grant, and I. J. S. Mann, "Green IT: An Implementation Framework," in *Proceedings of AMCIS*, 2009.

[14] K. Hoesch-Klohe, A. Ghose, and L.-S. Lê, "Towards green business process management," in *Proceedings of the 2010 IEEE International Conference on Services Computing*, ser. SCC '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 386–393. [Online]. Available: http://dx.doi.org/10.1109/SCC.2010.21

[15] N. Zenker and J. Rajub, "Resource Measurement for Services in a heterogeneous Environment," in *Proc. of ICTTA*. Damascus, Syria: IEEE Communications Society, 2008, pp. 1–15.

[16] D. Borgetto, G. Da Costa, J.-M. Pierson, and A. Sayah, "Energy-aware resource allocation," in *Proc. of Int'l Conference on Grid Computing*, October 2009, pp. 183–188.

[17] R. Nathuji, C. Isci, and E. Gorbatov, "Exploiting platform heterogeneity for power efficient data centers," in *Proceedings of the Fourth Int'l Conference on Autonomic Computing*. Washington, DC, USA: IEEE Computer Society, 2007, p. 5.

[18] D. Ardagna, M. Tanelli, M. Lovera, and L. Zhang, "Black-box performance models for virtualized web service applications," in *Proc. of WOSP/SIPEW*. New York, NY, USA: ACM, 2010, pp. 153–164.

[19] D. Ardagna, C. Cappiello, M. Lovera, B. Pernici, and M. Tanelli, "Active energy-aware management of business-process based applications," in *Proc. of ServiceWave*, 2008.

[20] A. M. Ferreira, K. Kritikos, and B. Pernici, "Energy-aware design of service-based applications," in *Proc. of ICSOC/ServiceWave*, ser. Lecture Notes in Computer Science, L. Baresi, C.-H. Chi, and J. Suzuki, Eds., vol. 5900, 2009, pp. 99–114.