Determining Power Consumption of Business Processes and their Activities to Enable Green Business Process Reengineering

Alexander Nowak, Tobias Binz, Frank Leymann

Institute of Architecture of Application Systems,
University of Stuttgart, Germany
{nowak, binz, leymann}@iaas.uni-stuttgart.de

Nicolas Urbach

Drees und Sommer AG,
Stuttgart, Germany
Nicolas.urbach@dreso.com

BibTeX:

```
@inproceedings{NowakBLU13,
  author    = {Alexander Nowak and Tobias Binz and Frank Leymann and Nicolas
               Urbach},
  title     = {Determining Power Consumption of Business Processes and their
               Activities to Enable Green Business Process Reengineering},
  booktitle = {Proceedings of the 17th IEEE International EDOC Conference,
               EDOC 2013, 09-13 September 2013, Vancouver, BC, Canada},
  year      = {2013},
  pages     = {259--266},
  doi       = {DOI 10.1109/EDOC.2013.36},
  publisher = {IEEE Computer Society}
}
```

Determining Power Consumption of Business Processes and their Activities to Enable Green Business Process Reengineering

Alexander Nowak, Tobias Binz, Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart
Stuttgart, Germany
firstname.lastname@iaas.uni-stuttgart.de

Nicolas Urbach
Drees und Sommer AG
Stuttgart, Germany
nicolas.urbach@dreso.com

Abstract— **Knowing and optimizing the environmental impact of business processes is gaining momentum for today's organizations. However, there is a lack of solutions that guide and support organizations in determining the power consumption of automated business processes, considering the resources and services used by that process. In this work we propose a comprehensive, multi-phased methodology and corresponding solutions that guide stakeholders through the methodology. As a proof-of-concept we build up an experimental testbed capturing the power consumption of Web services and propagating this information to business processes. The proposed approach supports stakeholders analyzing their automated business processes with respect to their environmental impact and is therefore supporting green business process reengineering.**

Keywords-**Business Processes, Green Business Process Reengineering, Green Business Process Management, Power Consumption, Enterprise Topology, Environmental Impact, Web Service, Key Ecological Indicators.**

I. INTRODUCTION

Today, the way organizations are doing business is widely represented and implemented as business processes [1]. Business processes compose and orchestrate different business services to achieve higher business objectives. Due to the fact that business processes nowadays are getting more and more complex [2], their optimization has become an inherent part of today's organizations. Besides the familiar optimization of business processes based on cost, quality, time, and flexibility aspects, Business Process Management (BPM) increasingly has to deal with new cross-cutting concerns, like compliance regulations [3] or the environmental impact of processes [4]. Trend surveys in power consumption of Information and Communication Technologies (ICT) [5], for example, point out that the power consumption of servers and mobile communication systems increases 16% to 20% per year. Such facts have led to an increasing awareness of customers, legislative authorities, and the general public towards sustainability. As a result, this trend motivates an increasing number of organizations to keep track of and improve their environmental impact in order to keep or extend their market shares [6][7].

A common but weak starting point is the use of resources with less environmental impact that can be easily put into place. However, these approaches are limited with respect to optimizations that may be achieved by improving the efficiency of how resources are used. Thus, the holistic optimization of business processes is one approach to deal with this issue as it combines both facets: *how* to use *what* kind of resources. In our previous work [8] we proposed an abstract approach to improve the environmental impact of business processes. We identified a set of patters describing different ecologically-aware optimization strategies, however, we did not emphasize how the environmental impact, and in the context of IT environments the power consumption in particular, of specific services implementing a business process can be determined. The problem addressed in this paper is therefore twofold: (i) Observing only the underlying computing resources might not give enough insight into the business operations using those resources, and (ii) observing only the business process activities might not indicate the corresponding power consumption as it is not clear which resources are used with which intensity. These problems indicate the need for a new approach that is able to handle the complete stack from business processes to software components and their underlying hardware resources. Moreover, when focusing on IT-resources and process-based applications, as we do in this work, the power consumption can only be measured at the hardware layer and, thus, we need proper techniques to propagate this information to the business processes based on their fraction of total power consumption.

The research questions of this work can be formulated as follows: (i) how does a measurement model for estimating the power consumption of Web services look like? (ii) What are the metrics that allow identifying power consumption of Web services? (iii) How may the power consumption of IT hardware resources be propagated through the application stack to the business process? To address those research questions our approach presents a methodology on how to determine the power consumption of activities of service-based business processes. To our best knowledge, there are no other approaches covering this complete area but only some parts thereof. However, unlike those approaches, e.g. [9][10][11][12], which are requiring highly specific and customized environments for measuring the power consumption and are, therefore, not usable in operative business, we present a holistic end-to-end methodology considering both existing and new methods. We do not require any specific hard- and software configuration and aim at the comprehensive support of decision makers that plan to optimize their business processes. The contributions of this paper can be defined as following: (i) a multi-phased methodology that guides stakeholders to identify the power consumption of activities of business processes, (ii) an approach for determining the power consumption along the underlying stack of hardware and software components of an automated business process, and (iii) a concept for aggregating and propagating environmental information from hardware to software to business processes considering various requirements and scenarios. As a proof-of-concept we provide an experimental testbed, as well as the corresponding measurement results.

The remainder of this paper is structured as follows: Section II describes the proposed end-to-end methodology that guides stakeholders to determine the power consumption of their business processes. Section III presents a possible solution on how to identify the hardware resources used by a business process. We also introduce a motivating example showing how this approach works. The definition of environmental metrics as well as a monitoring model for identifying the power consumptions of Web services is described in Section IV. Based on that information, Section V describes a concept for propagating the identified energy consumption from hardware to process level. Subsequently, Section VI presents an experimental testbed as proof of concept, Section VII presents the related works, and Section VIII closes the paper with a conclusion and future work.

II. AN END-TO-END METHODOLOGY

In this section we describe our methodology that guides stakeholders to determine the power consumption of business processes-based applications. An overview is given in Fig. 1. The methodology addresses two types of stakeholders: management decision makers that are interested in the environmental impact of their business processes and the engineers or developers that are responsible for the realization of optimization approaches. In the first step the management stakeholders need to specify the processes of interest and define the corresponding Key Ecological Indicators (KEIs) [13] to derive different metrics and measurement values. In general, KEIs can be very diverse as they cover a wide range of environmental indicators: water consumption, CO2 emission, soil pollution, hazardous liquids used for production, etc. Within this work, however, we only consider automated business processes performed on computing resources and, therefore, focus on how to determine the power consumption of those resources. The proposed methods we use here are described in Section IV in detail. Please note that this is not a limitation of our approach rather than describing one possible application scenario. For example, if there are human activities involved in the business process they can also be annotated manually to the corresponding activities.

After choosing a business processes the engineers need to identify the supporting infrastructure including the underlying soft- and hardware of a given business process and its services. To identify the underlying infrastructure and the dependencies to the business processes we propose to use an approach from our earlier work, presented in [14]. This approach uses a graph-based representation of the IT topology of an enterprise and allows performing different operations on this graph. Moreover, the application of multiple operations can be combined into strategies.

For the approach presented in this paper we use the workflow-deep-dive strategy that is intended to identify all relevant topology components that are underneath a specific business process. Details are presented in Section III. By knowing the corresponding soft- and hardware components, the values of the specified metrics are derived by using measurement facilities and calculation models. Whenever all necessary measurement information is available, the final step is to aggregate and propagate the metric values in reverse order up to the services and the business process activities, considering their individual fraction of the total power consumption. This step is described in Section V.

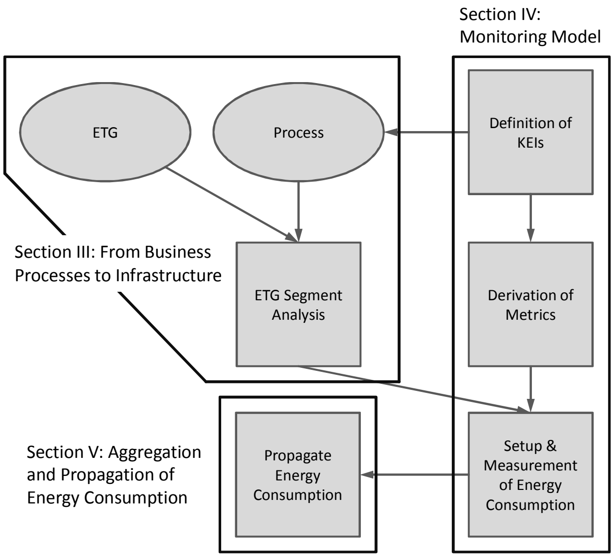

Fig. 1. Overview of our methodology

To realize the proposed methodology within an organization, we provide an adequate set of methods that can be used for each of these phases. A detailed description of those methods is presented in the following sections.

III. FROM BUSINESS PROCESSES TO INFRASTRUCTURE

We understand business processes as a composition of different self-contained services which fulfill a certain business goal. These services are performed using a variety of IT resources and, therefore, the proper identification of the IT-topology of organizations is an important aspect. In the following sections we describe a method on how to utilize Enterprise Topology Graphs (ETGs) [14] to identify the IT-topology that belongs to a business process, followed by an example that is later on used to show the feasibility of our approach.

A. Business Processes and their Infrastructure

Business processes of enterprises can be described in many different ways [15]. In the scope of this work we address automated business processes written in languages like BPEL [16] and BPMN [17]. Our goal is to analyze the environmental impact of those business processes, which includes the services they are orchestrating, as well as the software and hardware implementing the services and operating the processes. This holistic approach is required because the environmental impact of business processes is bound to the properties of the respective services and can only be optimized to some degree on the business process level. The environmental impact of the services in turn is mainly determined by the infrastructure operating them.

An ETG is a snapshot of the enterprise IT, overarching all layers from infrastructure to software and their relations. To represent this information in a suitable way, we have defined a graph-based model to represent and operate on ETGs [14]. Those graphs may be extracted from existing documentation, service descriptions, or manually. Each entity of the enterprise topology is represented as typed node which is connected by typed and directed edges. The graph-representation allows to apply existing and well proven graph algorithms to IT management problems. In [18] we have further defined a number of operations on ETGs, which can be combined into higher level analysis strategies. The workflow deep-dive strategy extracts for a given business process a topology segment, containing all nodes in the ETG which are needed to run this business process. This is the workflow management system itself, all services orchestrated in the business process, and their implementation, respectively.

B. Motivating Example and Use Case

To better illustrate how our approach can be applied we use an Extraction-Transformation-Load (ETL) process of a sales company that creates and prepares data for a data warehouse. This automated business process consists of three activities. The first one extracts various data from different databases that belong to different departments. The second activity ensures proper quality of that data by removing duplicates in customer data, for example. After that, the third activity loads the cleaned data to the data warehouse of the company. Fig. 2 presents the business process as BPMN model and the corresponding topology graph, i.e. the ETG segment, of the infrastructure.

The implementation of each of the activities is encapsulated as Web service and hosted on an application server. The Web services are also connected to a database, indicated by the ‘uses’ relation. Both, the database and the application server are hosted on a Windows Server 2008 R2 platform that runs on a HP ProLiant DL360 G6 server system. In this scenario the software stack running on the server is only a ‘blackbox’ with respect to power consumption. In the following sections we provide a set of possible solutions that can be used for determining the power consumption of each activity and subsequently of the complete business process.

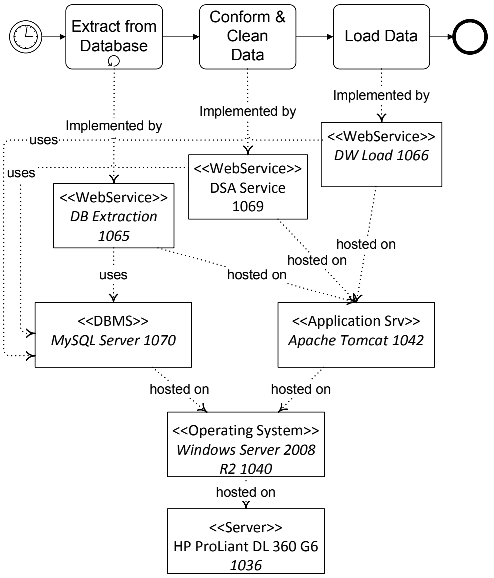

Fig. 2. Business process and corresponding infrastructure

IV. A MONITORING MODEL FOR IDENTIFYING THE POWER CONSUMPTION OF WEB SERVICES

In this section we propose a calculation model that enables us to determine the power consumption of Web services. Consequently, the ecological characteristic metric is defined as the power consumption of a business process and its activities, denoted in Watt seconds (Ws). The calculation specification is described in the following sections while the target value function is subject to the individual strategic objectives of an organization. Contrary to our model existing approaches like [12] rely on specific monitoring environments to determine the power consumption of IT systems. Introducing modified kernel versions of an operating system to gather specific hardware counters as well as board-level measurements might be more precise but are not applicable in operational business environments. To avoid this major constraint and enable the application in practice our framework solely uses standard OS-level performance counters. These high-level metrics are easy to collect and portable across different platforms. Therefore, there is no need to adapt existing systems or hardware.

Our model consists of two phases, a training phase and an application phase. Details on the training phase are presented in Section IV.A, details on the application phase in Section IV.B.

A. Identifying the Power Consumption based on Power Performance Models

Power performance models can be used to estimate the power consumption of a computing resource without the need of using a power meter. To build such a model we simultaneously measure the total power consumption and the load induced to the hardware components while performing different workloads. Depending on the utilization state of the system, represented by different performance counters like CPU utilization or hard disk usage, the power consumption will vary. We consider such a vector of different performance counters as a state of a system. TABLE I presents a sample set of performance counters. These performance counters are very abstract and are directly provided by the operating system, i.e., there is no need to modify the system. To get an accurate estimation of the power consumption of each utilization state it is recommended to use a variety of hardware performance counters capturing different components of the system.

TABLE I. HARDWARE PERFORMANCE COUNTERS

Hardware Components	**Performance Counter**	
CPU	Frequence [MHz]	CPU Time [%]
Memory	Read-Write operations [Pages/s]	
Hard disk	Read-Write operations [bytes/s]	
Network	Transfered Data [bytes/s]	

To identify the impact of each performance counter on the total power consumption we initially need to perform various training runs composed of different workloads that stress the system with varying utilization levels. The samples of these runs are used to correlate the power consumption with the utilization states of the hardware resources. Using a regular linear regression is one commonly applied approach for correlating the values and creating a corresponding model. However, both literature [19][20] and our own experiments have shown that the model resulting from a regular linear regression does not properly fit the measured values over the complete range of utilization states of the hardware. Therefore, we recommend using a switching model consisting of multiple linear regressions models. Moreover, we recommend using the CPU utilization as distribution key for the models as our experiments have shown that CPU is one of the most influencing factors for power consumption. Thus, the samples of the training runs used for the regression, i.e. the pairs of the measured system power consumption $\mathrm{P_{system}}$ and the selected hardware performance counters x_i, together denoted as TS, are chosen based on the corresponding CPU utilization of the system states while performing different Web services. Consequently, we create k different models for each subset ts_k from $TS = ts_1 \cup \ldots \cup ts_k$. Equation (1) describes the corresponding function for each regression model of the subsets ts_k as following: let P_{system} be the total power consumption measured at the system, let n be the number of samples of ts_k, let x_i be the vector of performance counters, let β_i be the vector of the impact weights of x_i, which we want to derive from the samples TS of the training runs, and let C be used as additional constant value. Then we define:

$$\mathrm{P_{system}} = \mathrm{C} + \sum_{i=1}^{n} \beta_i \mathrm{x}_i \quad (1)$$

The advantage of using this approach is twofold: (i) we measure the complete power consumption of a hardware resource by using a common power meter. We do not need any specific operating system adaptations or software measurement facilities. The only things we need to track are the various OS-level performance counters. (ii) After identifying the power consumption of the different states of a computing resource in the training phase there is no need of using a power meter anymore. The power consumption can be directly derived from the state of the hardware while performing different tasks.

B. Estimating the Power Consumption of a Service Request

As described in the previous section, the power consumption of a computing resource depends on the individual utilization state of the resource within a given time period. The major challenge is that each state of a computing resource, i.e. each individual workload that is evoked by a service, leads to another instantaneous power consumption. To get the exact instantaneous power consumption, continuous measurement facilities would be necessary. This is, however, very cumbersome and would require very specific measurement devices. As an approximation we are using the training phase to determine the average instantaneous power consumption of the services running on the system. To identify the power consumption of a single request, in the following denoted as $P_{service}$, we first identify the average instantaneous power consumption of this service solely running on a system. Furthermore, we repeat this measurement for all other service running on that system to identify the respective instantaneous power consumption, in the following subsumed as $P_{otherServices}$. Subsequently, we use the ratio between those values to proportionally distribute the total instantaneous power consumption of a system between the different services running on that system. Note, P_{total} also includes the instantaneous power consumption of the system in idle mode. Thus, we distribute P_{idle} between all services running on the system based on the ratio of their individual power consumption. Moreover, using P_{total} also allows us to distribute the power consumed by the corresponding middleware between the different services.

Let $PCR_{service}$ be the estimated average instantaneous power consumption of a service request and let $|req_{service}|$ be the number of service requests of the service to be evaluated. Using the total instantaneous power consumption P_{total} derived from our regression model and the ratio between the different services running on that system, we then define $PCR_{service}$ as:

$$\mathrm{PCR_{service}} = \frac{P_{total} * \frac{P_{service}}{P_{service} + P_{otherServices}}}{|req_{service}|} \quad (3)$$

V. AGGREGATION AND PROPAGATION OF POWER CONSUMPTION

The propagation of power consumption is used to determine the power consumption of each activity of a business process and consequently of the business process itself. However, we are not able to just use the power consumption of all service request because a service may be invoked multiple times within one and the same process instance or even from different business processes used in a process-based application. For example, in the event of having loops within a process model, the process model does not represent those activities multiple times. Thus, we need suitable propagation and aggregation mechanisms that allow to represent the power consumption metrics (i) of a specific process instance and (ii) of the process model within a given time period, i.e. a specific number of instances. On the business process side we therefore need to provide methods that (i) assign the power consumption of a service request to the corresponding activity and (ii) aggregate the power consumption of multiple executions or instances to the process model.

To address these aspects we propose a multi-phased approach. In the first step the power consumption of each service request needs to be determined or estimated. Details on this step are presented in Section IV. After that we need to distinguish between two scenarios depending on whether process monitoring information is already available or not. If the information is available we need to identify the number of service requests that are either tracked during process execution or gathered from process logs. If the monitoring information is not available we need to identify the new or modified parts of the process model, estimate the number of service requests and calculate the probabilities of invoking the service. If all information is available we can continue with calculating the power consumption for each activity, each process instance and the business process model. The different scenarios are described in detail in the following.

A. Case 1: Process Monitoring Information is available

In this section we want to describe how already completed process instances can be analyzed. Let $a \in A$ be an activity with A describing all activities of a business process model executed within a process instance $i \in I$, where I represents all instances of a business process model. Furthermore, let $|req_a|$ be the number of service requests of an activity a, let PCR_a be the average instantaneous power consumption per request of the service invoked by activity a, let Ψ be a weighting factor that allows to consider a varying power consumption per request, and let $T = \frac{1}{n}\sum_{i=1}^{n} t_i$ be the average duration of activity a. We then define the power consumption of an activity a of a process instance i as:

$$PC_i^a = |req_a| * PCR_a * \Psi * T \tag{4}$$

Please note that for this work we have assumed that each service request consumes the same amount of power, i.e. $\Psi = 1$. Based on that assumption we also assume that the duration of an activity does not vary significantly and we can use the average execution time of all activity instances. To calculate the power consumption PC_i of a complete process instance $i \in I$ let $j = (1, \ldots, m)$ be number of activities of a business process model and let E be the power consumption of the process engine performing this instance. We then define PC_i as:

$$PC_i = \sum_{j=1}^{m} (|req_j| * PCR_j * \Psi * T) + E \tag{5}$$

To support comprehensive analysis of process models, comprising multiple process instances, we also need to provide aggregated information for a specific number of process instances. Therefore, let $ii \subseteq I$ be a subset of process instances, let Φ be an aggregation function, and let $|ii|$ be the number of process instances. We then define the aggregated power consumption of a process model, PC_{total}, for using different aggregation functions Φ on the process instance level, as:

$$PC_{total} = \Phi\{PC_{ii(1)}, PC_{ii(2)}, \ldots, PC_{ii(|ii|)}\} \tag{6}$$

Furthermore, we define PC_{total}^a for using different aggregation functions Φ on the activity level of a single process instance as:

$$PC_{total}^a = \Phi\{PC_{ii(1)}^a, PC_{ii(2)}^a, \ldots, PC_{ii(|ii|)}^a\} \tag{7}$$

This scenario may be useful if certain activities are executed multiple times within one process instance. The aggregation functions Φ can be the sum, average, minimum, or maximum, for instance.

B. Case 2: Monitoring Information is not/partly available

In this section we consider the analysis of modified or new business process models where no measurement information is available. In this case we first need to identify the modified or new parts of the business process. If the business process model has only been changed in parts we can use the available monitoring information for the remaining activities. Subsequently, we need to determine the probabilities of the different paths of the process model that may be executed during runtime [21][22]. Let p_f^a be the probability of following the control flow towards activity a at fork f with $f = 1..n$ representing the number of forks in the control flow. We then define the probability of executing an activity a, p^a as

$$p^a = \prod_{f=1}^{n} p_f^a \tag{8}$$

Thus, p^a depends on the existing forks and joins in the execution flow of the process model and is calculated as product of the individual probability of each fork f. Using an estimated number of service requests we then define the power consumption of an activity a as:

$$PC'^a_i = p^a(|req_a| * PCR_a * \Psi * T) \tag{9}$$

The power consumption of the complete process instance is defined as:

$$PC'_i = \sum_{j=1}^{m} (PC'^a_i(j)) + E \tag{10}$$

Note, that the aggregation and propagation model proposed in Section V considers the function E for representing the power consumption of the process engine that is performing a business process. Previous work [23] has shown that the power consumption of the process engine may influence the total power consumption significantly, especially when performing distributed business processes. Thus, we use E in this work more as a placeholder and want to build and formalize a separate model in future work.

VI. EXPERIMENTAL TESTBED

Our experiments have been performed on a HP ProLiant DL360 G6 running Windows Server 2008 R2. For measuring the power consumption of the system we used a standard Watts Up! Pro power meter. The used performance counters are provided natively by the operating system and are described in TABLE I. As a real world example we are using the ETL process described in the motivating example in Section III.B. The application stack is setup as shown in Fig. 2. The process consists of three activities each implemented as a single Web service. The goal of this scenario is to determine the average power consumption of a sample set of process instances in order to evaluate the environmental impact of that business process model. Thus, we have defined the ecological characteristic metric for each activity of the process as the power consumption. As described in Section V we omit the power consumption of the process engine at this point in time.

For the training phase we used different synthetic Web services and various micro-benchmarks to simulate different utilization states of the system. The synthetic Web services calculated prime numbers, generated random strings, downloaded files from other servers and wrote random files to the hard disk. The micro-benchmarks we used are LoadTester, StressMyPC, MemAlloc, MyCPU. Subsequently, we correlated the different utilization states and the corresponding measured power consumption using the power performance model proposed in Section IV.A. Our experiments have shown that CPU time is the predominant influence factor for total power consumption. Therefore, we chose the CPU time as distribution key for creating the different regression models. Based on the observations of the samples of the training runs we decided to separate them into four different subsets $TS = ts_1 \cup \ldots \cup ts_4$. The different subsets are dividing the training samples based on the corresponding CPU utilization (see Equation 11). As a result, we get four different linear regression models $M_1 \; \ldots \; M_4$.

$$M = \begin{cases} M_1 \;\; for \;\; 0 < x_{CPU} \leq 10 \\ M_2 \;\; for \;\; 10 < x_{CPU} \leq 20 \\ M_3 \;\; for \;\; 20 < x_{CPU} \leq 30 \\ M_4 \;\; for \;\; 30 < x_{CPU} \leq 100 \end{cases} \tag{11}$$

Each of those models is created by using Equation (1), considering the corresponding sample subset ts_k. The linear regression for each model allows us to derive different influence factors for each model, i.e. the weightings of each indicator to the total power consumption of the system. In the next step of our experiment we performed the ETL process from the sample scenario multiple times without and with varying background workloads, i.e. other services running on the system. The experiment was implemented using the steps presented in TABLE II. To determine the accuracy of our model, we also used the measurement device for each step to capture the actual power consumption of the system. Later on we use these values to compare them with the ones estimated from our model.

TABLE II. EXPERIMENT SETTINGS

	Steps of the experiment - application phase
1	While System is on idle
2	Capture $\mathrm{P_{total}}$ and x_i
3	For each Web Service Request from ETL Process
4	Capture $\mathrm{P_{total}}$ and x_i
5	For each Background Web Service and Load
6	Capture $\mathrm{P_{total}}$ and x_i
7	For each Web Service Request from ETL Process while performing Background Web Service
8	Capture $\mathrm{P_{total}}$ and x_i

In the first step of our experiment we set the system in idle mode and captured the power consumption as well as the different performance counters x_i. In this step, $\mathrm{P_{total}}$ equals $\mathrm{P_{idle}}$. After the stabilization of the system, we performed only the ETL process, i.e. the Web services implementing this process, and captured $\mathrm{P_{total}}$ and x_i again. After that, the same step was performed for a second Web services from our synthetic Web services. In the final step, we performed the ETL process, i.e. the Web services implementing this process, while running the second service simultaneously. And again, we captured $\mathrm{P_{total}}$ and x_i.

TABLE III. INSTANTANEOUS POWER CONSUMPTION PER WEB SERVICE REQUEST

	Extraction			Transformation			Load		
	PCR$_m$ (W)	PCR$_e$ (W)	δ (%)	PCR$_m$ (W)	PCR$_e$ (W)	δ (%)	PCR$_m$ (W)	PCR$_e$ (W)	δ (%)
PI$_1$	58,02	59,94	*3,30*	62,78	62,68	*-0,17*	66,07	64,55	*-2,30*
PI$_2$	65,22	64,17	*-1,61*	63,70	62,63	*-1,67*	65,26	64,90	*-0,56*
PI$_3$	65,20	63,99	*-1,85*	63,56	62,64	*-1,45*	66,83	64,43	*-3,59*
PI$_4$	65,36	61,80	*-5,44*	63,06	62,79	*-0,43*	67,29	66,01	*-1,91*
PI$_5$	65,24	63,99	*-2,19*	62,66	62,67	*0,02*	66,42	64,87	*-3,00*
PI$_6$	65,66	64,11	*-2,36*	63,36	62,59	*-1,22*	65,65	64,89	*-1,17*
AVG	64,13	63,00	*-1,76*	63,36	62,67	*-1,20*	66,29	64,87	*-2,15*

Using the model proposed in Section IV.B we were now able calculate the instantaneous power consumption of each Web service that has been invoked within the ETL process while some other services running on that system. The complete results for all Web service requests used in six sample instances of the ETL process are shown in TABLE III. Here, we also provide the results calculated from the measured values in order to evaluate the accuracy of our model against those values. PCR$_m$ represents the measured values and PCR$_e$ represents the estimated values from our model. We can now use the estimated instantaneous power consumption of each Web service request and propagate this information to the ETL business process using these Web services. Recall, that we are using the number of service requests to initially propagate the power consumption to the process level. In our example scenario each Web service is called one after the other and only one time within each process instance execution. Thus, we have $|req| = 1$ for all Web service requests. Also remember, that we assume that each service request consumes the same amount of power, i.e. $\Psi = 1$, that we omit the power consumption of the process engine, i.e. $E = 0$, and that we use the average activity duration to calculate the total power consumption.

Based on the proposed model presented in Section V we are now calculating the average power consumption of the proposed ETL process considering multiple process instances. In the first step we are using Equation (4) to calculate the power consumption for each activity a of each process instance i, i.e. PC_i^a. For T we have used the average duration of the activities, i.e. $T = \frac{1}{n}\sum_{i=1}^{n} t_i$. The complete results for all activities are shown in the center part of TABLE IV.

TABLE IV. AVERAGE POWER CONSUMPTION OF ETL PROCESS

	$PC_i^{Extraction}$ in (Ws)	$PC_i^{Transform.}$ in (Ws)	PC_i^{Load} in (Ws)	PC_i in (Ws)
Duration	9,97 s	16,50 s	6,62 s	-
PI_1	597,74	1034,28	427,50	2059,52
PI_2	636,97	1033,55	429,79	2103,32
PI_3	638,21	1033,63	426,72	2098,56
PI_4	616,40	1036,16	437,13	2089,69
PI_5	638,16	1034,14	426,70	2099,00
PI_6	639,33	1032,88	429,71	2101,91
	$PC_{total}^{Extraction}$	$PC_{total}^{Transform}$	PC_{total}^{Load}	PC_{total}
$\Phi = \mathbf{AVG}$	628,30	1034,10	429,59	2092,00

To get the power consumption of each instance, i.e. PC_i , we sum up the individual power consumption of all activities as shown in Equation (5). These values are presented in the right-hand column of TABLE IV and can be used to evaluate single instances of the process. To provide aggregated information we are using Equation (6) and (7) to calculate the average power consumption of all six process instances on activity and instance level, respectively. The results, i.e. PC_{total}^{a} and PC_{total}, are presented on the bottom row of TABLE IV. PC_{total} = 2092,00 Ws indicates, that performing the ETL process of our example scenario and on our system consumes 2092,00 Ws in average. The involved Web services consume 628,30 Ws, 1034,10 Ws, and 429,59 Ws, respectively. This information can be used by management stakeholders as an indicator for the environmental impact of this business processes and is the basis for environmental optimization decisions and strategies.

VII. RELATED WORK

Available literature and approaches already cover a wide range of individual solutions for capturing the energy consumption of IT systems or applications. Their focus, however, is very limited to either specific domains or specific hardware and software requirements. The approach of Ardagna et al. [24] deals with energy-aware resource allocation mechanisms and policies for service oriented architectures. Capiello et al. [25] have extended this approach to enable the definition of service alternatives during design time to address the tradeoff between service performance and power consumption. Nowak et al. [8] have captured a set of patterns that help process designers to improve the environmental impact of business processes. However, all of these approaches have not covered aspects of imposing the energy or environmental information. This information is, however, vital to any optimization approaches.

To identify the power consumption of IT resources different energy profiling approaches exist. For the most parts of those approaches the correlation of power consumption and hardware utilization or application throughput has been used to estimate the power consumed by an application. Hi Chen et al. [26] have developed a tool to estimate the power consumption induced by a process by converting performance events into power consumption. Koller et al. [27] proposed an application aware power meter called WattApp which calculates the power consumption of an application as a function of its throughput. Xian et al. [10] use a modified Linux kernel to assign power samples to programs. Kansal and Zhao [11] also use windows event tracing to apply learning algorithms that predict the power consumption of virtual machines and support application designers during design time. Do et al. [9] present a software-based approach that collects power consumption data from all applications based on a modified Linux kernel.

In our work, we have extended the basic ideas of these approaches for determining the power consumption of business process-based applications in a widely applicable fashion. Unlike our approach, the objective of the majority of the previous works is solely to estimate the power consumption of specific hard- and software systems without taking the corresponding business processes into account. Moreover most of these works are based on specific operating systems, modified kernel versions and additional software tools which make the approaches not easily applicable and interoperable in practice.

VIII. CONCLUSION AND FUTURE WORK

To address the environmental impact of business process this work presents a multi-phased methodology that guides stakeholders through the process of identifying the resources of a business process, defining ecological characteristic metrics and their corresponding measurement and calculation specifications, and propagating this information up to the business process level. The work also presents possible solutions for each of these phases focusing on the energy consumption of business processes. It describes how Key Ecological Indicators can be used to define the environmental impact, how power performance models can be used to identify the energy consumption of single Web services, and how the information from hardware resources can be propagated to the business processes. So far, the proposed solution does not cover every aspect in detail, in particular the consideration of external services and the energy consumption of possibly distributed process engines. Nevertheless, we argue that the proposed methodology and its solutions are a good starting point for future development.

Therefore, in our future work we plan to extend the proposed methodology in different ways. First, we want to develop a suitable model for considering the energy consumption of the process engine. Second, we want to rebuild the model concerning the constant energy consumption per Web service request. Using distribution functions should give us more precise values. Another major issue we want to address is the changing energy consumption depending on the server utilization. So far, the energy consumption is based on the basis utilization of the server. The energy consumption is therefore equally distributed between the total number of requests. Although this is a good approximation, we might achieve a more precise result if we can identify the overlap of different service requests within a specific time period, i.e. whether the services are executed in sequence, parallel, or something in between.

ACKNOWLEDGMENT

This work was partially funded by the BMWi projects CloudCycle (01MD11023) and Migrate! (01ME11055). We also thank Drees&Sommer for their support.

REFERENCES

[1] M. Weske, Business Process Management: Concepts, Languages, Architectures. Springer, 2007.

[2] J. Vanhatalo, H. Völzer, F. Leymann, Faster and more focused Control-flow Analysis for Business Process Models though SESE Decomposition. Proc. of ICSOC 2007, Springer, 2007, pp. 43-55.

[3] D. Schleicher et al., Compliance Domains: A Means to Model Data-Restrictions in Cloud Environments. Proc. of EDOC 2011, IEEE, 2011.

[4] A. Nowak, F. Leymann, D. Schumm, The Differences and Commonalities between Green and Conventional Business Process Management. Proc. of CGC 2011, IEEE, 2011.

[5] G. Fettweis, E. Zimmermann, ICT Energy Consumption - Trends and Challenges Communications. Proc. of WPMC 2008, IEEE, 2009.

[6] Gartner Research, "Gartner Identifies the Top 10 Strategic Technologies for 2010", available at http://www.gartner.com/it/page.jsp?id=1210613, 2010.

[7] GreenBiz.com, "How Will Sustainability Change at Your Company in 2011", available at http://www.greenbiz.com/blog/2010/12/29/how-will-sustainability-change-your-company-2011, 2011.

[8] A. Nowak, F. Leymann, D. Schleicher, D. Schumm, S. Wagner, Green Business Process Patterns. Proc. of PLoP 2011, ACM Digital Library, 2012.

[9] T. Do, S. Rawshdeh, W. Shi, pTop: A Process-level Power Profiling Tool. Proc. of HotPower '09, ACM, 2009.

[10] C. Xian, L. Cai, Y.-H. Lu, Power Measurement of Software Programs on Computers With Multiple I/O Components. IEEE Transactions on Instrumentation and Measurement, 2007, 56(3), pp. 2079 -2086.

[11] A. Kansal, F. Zhao, Fine-grained energy profiling for power-aware application design. SIGMETRICS Perform. Eval. Rev., ACM, 2008, 36, pp. 26-31.

[12] P. Bartalos, M.B. Blake, Green Web Services: Modeling and Estimating Power Consumption of Web Services. Proc. of ICWS 2012, IEEE, 2012, pp. 178-185.

[13] A. Nowak, F. Leymann, D. Schumm and B. Wetzstein, An Architecture and Methodology for a Four-Phased Approach to Green BPR. Proc. of ICT-GLOW 2011, Springer, 2011, pp. 150-164.

[14] T. Binz, C. Fehling, F. Leymann, A. Nowak, D. Schumm, Formalizing the Cloud through Enterprise Topology Graphs. Proc. of CLOUD 2012, IEEE, 2012, pp. 742-749.

[15] F. Leymann, D. Roller, Production Workflow - Concepts and Techniques. Prentice Hall PTR, 2000.

[16] OASIS, Web Service Business Process Execution Language Version 2.0. OASIS, 2007.

[17] OMG, Business Process Model and Notation (BPMN). OMG, 2012.

[18] T. Binz, F. Leymann, A. Nowak, D. Schumm, Improving the Manageability of Enterprise Topologies Through Segmentation, Graph Transformation, and Analysis Strategies. Proc. of EDOC 2012, IEEE, 2012, pp. 61-70.

[19] P. Bartalos, M.B. Blake, S. Remy, Green Web Services: Models for energy-aware web services and applications. Proc. of SOCA 2011, IEEE, pp. 1-8.

[20] J.D. Davis, S. Rivoire, M. Goldszmidt, E.K. Ardestani, No Hardware Required: Building and Validating Composable Highly Accurate OS-based Power Models. Microsoft Technical Report, 2011.

[21] F. Rosenkranz, Geschäftsprozesse: Modell- und computergestützte Planung. 2. Edition, Springer, 2005.

[22] K. Hinge, A. Ghose, G. Koliadis, Process seer: A tool for semantic effect annotation of business process models. Proc. of EDOC 2009, IEEE, 2009, pp. 54-63.

[23] S. Wagner, D. Roller, O. Kopp, T. Unger, F. Leymann, Performance Optimizations for Interacting Business Processes. Proc. of IC2E 2013, IEEE, 2013.

[24] D. Ardagna, C. Cappiello, M. Lovera, B. Pernici, M. Tanelli, Active Energy-Aware Management of Business-Process Based Applications. Proc. of ServiceWave 2008, Springer Berlin-Heidelberg, 2008, pp. 183-195.

[25] C. Cappiello, M. Fugini, G. Gangadharan, A. Ferreira, B. Pernici, P. Plebani, First-Step toward Energy-Aware Adaptive Business Processes On the Move to Meaningful Internet Systems. Proc. of OTM 2010 Workshops, Springer Berlin-Heidelberg, 2010, pp. 6-7.

[26] H. Chen, Y. Li, W. Shi, Fine-Grained Power Management Using Process-level Profiling. Sustainable Computing: Informatics and Systems, 2012, 2(1), pp. 33-42.

[27] R. Koller, A. Verma, A. Neogi, WattApp: An Application Aware Power Meter for Shared Data Centers. Proc. of ICAC 2010, ACM Ney York, 2010, pp. 31-40.