Automating Green Patterns to Compensate CO_2 Emissions of Cloud-based Business Processes

Alexander Nowak, Uwe Breitenbücher, Frank Leymann
Institute of Architecture of Application Systems (IAAS)
University of Stuttgart
Stuttgart, Germany
firstname.lastname@iaas.uni-stuttgart.de

***Abstract*—The usefulness of patterns to optimize the environmental impact of business processes and their infrastructure has already been described in literature. However, due to the abstract description of pattern solutions, the individual application of patterns has to be done manually which is time consuming, complex, and error-prone. In this work, we show how the Green Compensation pattern can be applied automatically to different individual Cloud-based business processes in order to lower the negative environmental impact of the employed Virtual Machines without any manual effort. We show how our Management Planlet Framework can be used to implement this concrete refined pattern solution in a reusable way.**

Keywords-Green Business Process Patterns; Management Automation; Cloud Computing; Infrastructure as a Service.

I. INTRODUCTION

Today, organizations widely use business processes to describe the way they are doing business. Business processes compose and orchestrate various business services that are used to reach certain business objectives. The management and optimization of these business processes and their supporting infrastructure, therefore, has become an inherent part of organizational development. So far, typical optimization dimensions have been limited to cost, time, quality, and flexibility. This scope has been extended in recent years by the consideration of ecological indicators and measurements [1][2]. Based on legal requirements and increased customer awareness, more and more organizations keep track of and improve their environmental impact to save or extend their market shares [3][4].

There have been developed first approaches to guide and support organizations with tracking their environmental impact, like the ISO Standard 14001 [5]. Beyond that, a fundamental concept to improve the environmental impact of business processes is based on patterns that capture abstract knowledge about solutions for frequently recurring ecological issues, forces, and challenges. In Nowak et al. [6][7], we presented a set of *Green Business Process Patterns* that capture expertise about how to optimize business processes in terms of their environmental impact. However, these pattern-based approaches typically need to be refined manually for individual use cases due to the generic nature of patterns. Thus, the abstract solution described by patterns needs to be refined towards the concrete individual application to which it is applied to, in this context, the actual business processes, services, and underlying infrastructures. The problem of this manual refinement step is that (i) a deep technical knowledge as well as profound application management expertise is required to map the described abstract solution concepts to concrete circumstances [12], (ii) manual transformation steps are typically error-prone and time consuming [8], and (iii) the integration of heterogeneous process and application infrastructures needs to be handled. Especially for complex business processes that are commonly employed in enterprises today, a manual application of patterns does also come with additional risks, namely (i) unintended changes of the processes' semantics, (ii) unpredictable side effects to other processes, and (iii) technical errors. Hence, the risk is much too high to apply such changes manually.

As a result, there is a gap between the abstract nature of patterns and their actual application to real use cases that prevents applying the concept of patterns efficiently to the domain of Green Business Process Management. In addition, when applying changes to Cloud-based applications invoked from business process, the changes must be automated to ensure Cloud properties such as self-service, on-demand computing, and elasticity [29]. Therefore, we need a means that allows applying the generic knowledge captured in Green Business Process Patterns automatically to individual use cases. In this paper, we tackle these issues and present a conceptual approach to apply a pattern called *Green Compensation* [6] generically to various kinds of IaaS-based business processes that employ Virtual Machines (VMs) as underlying infrastructure service. The contributions of this paper are threefold: we first (i) describe the *concrete* refinement of the *abstract* Green Compensation pattern towards this VM-based use case, (ii) we present how the refined pattern can be automated independently from individual applications using the Management Planlet Framework [9][10][11][12], and subsequently we describe(iii) how our realization enables Cloud providers as well as customers to decrease the ecological impact of their applications significantly without any manual effort.

The remainder is structured as follows: Section II describes background information about patterns and their application in general as well as the Green Compensation pattern and its use in a motivating scenario. Section III

 ISBN: 978-1-61208-354-4

describes our approach by refining the Green Compensation pattern solution and applying the methods from the employed Management Planlet Framework. Section IV provides related work and Section V concludes the paper.

II. Background, Requirements, and Motivation Scenario

In this section, we provide background information about patterns in general and the difficulties of automating their application. We introduce a Green Business Process Pattern named *Green Compensation* and a motivating scenario that is used throughout the paper explain our approach.

A. Patterns and their Application

Patterns describe problems that we can find over and over again in our environment or knowledge domain and the corresponding core of a solution to that problem [13]. Such an abstract solution, therefore, does not describe a single solution for a specific use case but describes the solution in a way such that it can be used again and again in individual scenarios [13]. Patterns usually have relations to other patterns indicating whether the fit together or not, or can be used as an alternative. Such a set of patterns for a particular domain is typically described as a *pattern language*.

To address ecological aspects of enterprises, we introduced *Green Business Process Patterns* [6][7] that help stakeholders to reduce the negative environmental impact of their business processes. Here, the term "green" is used as a synonym for all aspects related to the environmental impact such as CO_2 emissions, pollution, waste, etc. Considering Cloud-based business processes, the environmental impact of a process not only depends on the structure of that process but also on the services that are invoked. Therefore, Green Business Process Patterns do not focus on the control and data flow of processes only, but also on the resources that are used as well as the communication among them.

Due to the abstract description of pattern solutions, their application typically requires a *manual refinement* of the patterns towards the actual use case in which they shall be applied [12][14]. Thus, the abstract solution described by a pattern must be transformed into a description of concrete management tasks that have to be executed to apply the pattern. In the domain of Cloud Computing, this means dealing with low level technical issues such as orchestrating management APIs or executing configuration scripts [10]. Unfortunately, this is a technically complex, time consuming, and error prone challenge [8][12]. To tackle these issues, Cloud application management and, therefore, applying patterns in this domain must be automated. As a result, the two main issues that have to be tackled when applying the concept of patterns efficiently to the domain of Cloud Computing are (i) handling the technical complexity of pattern refinement for individual use cases and (ii) automating this refinement step and its actual execution to avoid manual errors [12]. The approach presented in this paper considers these issues and provides an automated means to apply patterns to individual use cases based on a formerly developed Application Management Framework.

B. Green Compensation Pattern

In this paper, we use a motivating scenario to better describe our approach. As part of this motivating scenario, we show how the pattern *Green Compensation* can be refined and applied in an automated manner in order to cover the complex management tasks necessary to realize this pattern. Therefore, we will first introduce the Green Compensation pattern by providing a shortened description of the pattern. The complete description of the Green Compensation pattern can be found in Nowak et al. [6].

To ease the use, handling, and identification of patterns, they are typically described in a common format. We use a shortened format originating from Nowak et al. [6] that consists of an *icon* and *intend* that allow a quick identification and classification of the pattern. Subsequently, the *context* describes the scenario in which the pattern might be used, the *problem* describes the forces the pattern is confronted with, the *solution* provides an abstract and implementation independent solution of the problem, the *result* describes the context after applying the pattern, and the *example* provides different scenarios where the pattern has been successfully applied to. Given that format, the Green Compensation pattern is described as follows:

Pattern: Green Compensation

Improve the negative environmental impact without changing the process model or the resources of a business process.

Context: An Enterprise uses a multi-services value stream and wants to contribute to the harmony and health of the environment without changing processes and resources.

Problem: Enterprises typically describe their business processes based on various functional requirements. This line of action ensures that business processes are able to reach defined business objectives. However, when considering environmental objectives, the design of a process may differ from the functional driven design. Due to internal or legislative guidelines, these changes in the process model may not be allowed. Thus, the challenge is to improve the negative environmental impact of a business process without changing its structure or employed resources.

Solution: Each time a business process or service that can't be altered is instantiated, a corresponding compensation process or activity will be instantiated as well. A compensation process or activity is used to compensate at least parts of the negative environmental impact by using services that invest in climate projects, for example. Therefore, the environmental impact must be quantified accordingly. Figure 1 describes the different steps of the abstract solution as a process modeled in BPMN [15].

Figure 1. Abstract Solution of the Green Compensation Pattern.

Result: Although the original processes or services will not be altered, their negative environmental impact can be reduced from a global point of view. The negative impact will not be eliminated, however, it will be compensated with a positive impact from other projects.

Example: The oil company JET offers customers in Germany a sustainability program that compensates the CO_2 emission caused by the amount of gas they refuel [16]. The railroad company Deutsche Bahn also offers a sustainability program where the CO_2 emissions of customers may be compensated [17]. Companies that do not run own programs can use the service of MyClimate [18], for example.

C. Motivating Scenario

In this section, we introduce a case study that is used as motivating scenario throughout the paper to illustrate the difficulties of refining abstract solutions from patterns and applying them in an automated fashion to individual use cases. Let us consider a company called *WiFo* that provides a service for electricity producers that forecasts the wind situation for the next day. For flexibility reasons, WiFo runs various service-based applications in a Cloud that are orchestrated by business processes. One of their main processes is depicted in Figure 2 and consists of three steps: (1) the simulation data needs to be set up, (2) the forecast calculation is performed, and (3) the results are analyzed.

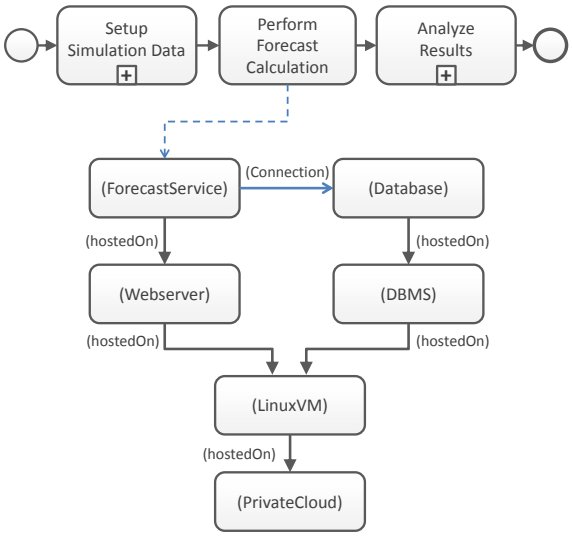

Figure 2. Motivating Scenario.

As we can see in Figure 2, step (1) and step (3) are modeled as sub-processes, i.e., these tasks include several other subtasks that are omitted here. Step (2) is modeled as single task that invokes a forecast service that is hosted on an Apache Tomcat running on an Ubuntu Linux operating system. The complete application stack is hosted on their private Cloud environment. To analyze their business processes, WiFo has made use of existing approaches in the domain of Green Business Process Management [19]. As a result of the analysis, WiFo decided to apply the Green Compensation pattern [6] as they do not want to change their process from a functional perspective, i.e., they do not want to change the control flow or employed resources.

For the application of the selected Green Compensation pattern, WiFo needs to take care of the following tasks. First, the negative environmental impact (often described as CO_2 equivalents) that is induced by the process needs to be quantified. The impact of using IT services depends mainly on the electricity consumption of the services and their corresponding infrastructure. In Nowak et al. [2], we investigated the main drivers for electricity consumption of a computer system and found that CPU utilization is the most dominant one. The challenge here is to (i) identify the infrastructure and all components that are employed a business process and (ii) to analyze which components are responsible for a negative environmental impact. In addition, (iii) these steps must be automated as argued in Section II.A.

The second challenging task is improving the environmental impact of the infrastructure and the employed components. Naively, WiFo could simply change the resources they are using. However, as WiFo does not want to change the employed resources and their configurations, we need some external means to improve the environmental impact without changing the environment. As described in the Green Compensation pattern, it might be suitable to use external CO_2 compensation services that offset the CO_2 emissions by supporting various climate projects. Thus, a suitable service for CO_2 compensation has to be found and an appropriate invocation of this service in the existing infrastructure has to be integrated based on CPU utilization

However, the integration of the CO_2 compensation service is not trivial as the infrastructure must be accessed to set up the service invocation. This typically requires a lot of technical expertise, for example, the login to a VM via SSH and the installation of the corresponding components. In addition, a monitoring agent that keeps track of the CPU utilization needs to be installed, configured, and connected to determine the CO_2 compensation of a service based on its utilization [2]. As an additionally aspect, all these steps must be performed rapidly to avoid system downtime. In summary, without having detailed knowledge about these steps, applying the abstract Green Compensation pattern results in a major management issue that must be executed carefully to (i) avoid system downtime caused by technical errors or slow execution and (ii) donating too much money caused by a wrong configuration that leads to false (e.g., too high) estimated CO_2 emissions. These results comply with the two issues described in Section II.A. Applying patterns in the domain of Cloud application management must, therefore, tackle (i) the technical complexity of solution refinement and (ii) the automation of refinement and solution execution. In this paper, we show how this can be realized using our approach for the Green Compensation pattern.

III. Automation of the Green Compensation Pattern for VM-based Business Processes

In this section, we present our approach to apply the Green Compensation pattern automatically to various kinds of VM-based applications that are hosted in the Cloud. We additionally show how the two requirements for applying

patterns in the domain of Cloud Computing, as introduced in Section II.A, are addressed and how the approach can be used to automate the application of the Green Compensation pattern to our motivating scenario described in Section II.C. This section is structured as follows: in the next subsection III.A, we first describe how CO_2 emissions of Virtual Machines can be determined and compensated. In section III.B, we refine the abstract solution of the Green Compensation pattern based on these compensation mechanisms towards an application for Virtual Machine-based business processes hosted in a Cloud. In section III.C, we show how the refined solution can be automated using our Pattern-based Management Planlet Framework we developed in former work. The result of this approach is a refined and fully-automated Green Compensation pattern that can be applied to various Virtual Machine-based applications for compensating their CO_2 emission.

A. Determination and Compensation of CO_2 Emissions caused by Virtual Machines

To apply the Green Compensation pattern to Virtual Machine-based service of a business processes, it is vital to identify their environmental impact. In Nowak et al. [2], we presented an approach that allows determining the energy consumption of a Web Service. In a nutshell, we distribute the total energy consumption of a physical system between the different services running on that system. As a means for distributing the total energy consumption, we are using the CPU utilization that is allocatable to each service. In case of having multiple VMs running on a physical system, we can extend that approach transitively and use the CPU utilization of a VM to distribute the total energy consumption of the hypervisor between the VMs running on that hypervisor.

To ease the use of external compensation services, we translate the energy consumption of a service and its underlying VM to CO_2 equivalents. To calculate the corresponding CO_2 equivalent from the energy consumption, the individual energy mix of a server location needs to be considered. In Germany, the Federal Ministry for Environment [21] provided an average of 0,55kg CO_2 per KWh in 2010. In the US, the average was about 0,59kg CO_2 per KWh [22] in 2009. Based on that conversion, various compensation services can be used such as PrimaKlima [21] or CarbonFund [25]. The main objective of those services is to offset CO_2-emissions through, for example, reforestation, investments in renewable energy, or the purchase of carbon credits that prevents businesses from pollution. We employ such compensation services to compensate the determined CO_2 emission of the Virtual Machines for which the measured negative environmental impact shall be improved.

B. A Method to refine the Green Compensation Pattern

All Green Business Process Patterns documented in Nowak et al. [6], including the Green Compensation Pattern shown in Section II.C, describe solutions in a very abstracted fashion. In order to apply them to automated business processes and their underlying IT-infrastructure, the given solution must be refined, i.e., the solution must be described as management steps that have to be performed in order to apply the pattern in the context of and focus on Virtual Machines. If we recall the motivating scenario from Section II.C, WiFo wants to compensate the "Perform Forecast Calculation" activity of their business process. This activity invokes a Cloud service that is performing the actual calculation. To compensate the negative environmental impact of the underlying Virtual Machine that is hosting that service, we need to describe a refined solution towards this VM-based context that covers all steps to be performed.

The Green Compensation pattern ensures that the changes to be applied do not change the actual processes and employed resources (cf. Section II.B). Therefore, to instantiate the compensation activity that improves the negative environmental impact by compensating CO_2 emission, this must be considered and the process as well as the resources must not be changed. As a result, we need a means to integrate this compensation activity *transparently* to the process. Therefore, we only install a new monitoring component on the Virtual Machine and do not change the other components or configurations of the process.

Figure 3 shows an overview of the method we applied to refine the pattern as BPMN diagram [15]. In the first step, the Virtual Machines that are responsible for the negative environmental impact of an activity need to be identified. Using that information, the next step includes the installation of a monitoring agent. In the BMWi funded project "Migrate!" [24], we have developed such an agent that is based on *sigar* [23]. That agent is able to gather the relevant CPU utilization of a VM. Details on which other performance counters should be tracked and on what time interval can be set in the following "Configure Monitoring Agent task". After choosing a Compensation Service, the Monitoring Agent needs to be connected to that service in order to initiate the compensation based on the actual environmental impact of the service. Here, different rules may be specified and configured depending on the business objectives. If WiFo wants to compensate, for example, 100 percent of their forecast service, the compensation service initiates a compensation for 100 percent of the total energy consumption of the service. In case the compensation should additionally be based on other aspects, like the number of invokes or the total environmental impact of WiFo, additional components need to be provided and integrated.

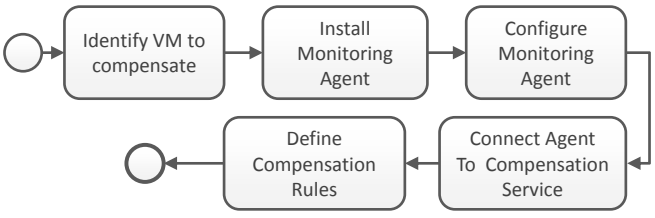

Figure 3. Refined Green Compensation pattern solution for Virtual Machine-based business processes.

C. Employed Management Framework

The employed Management Planlet Framework [9][10] [11][12] enables the automation of applying management patterns by generating declarative models that can be briefly

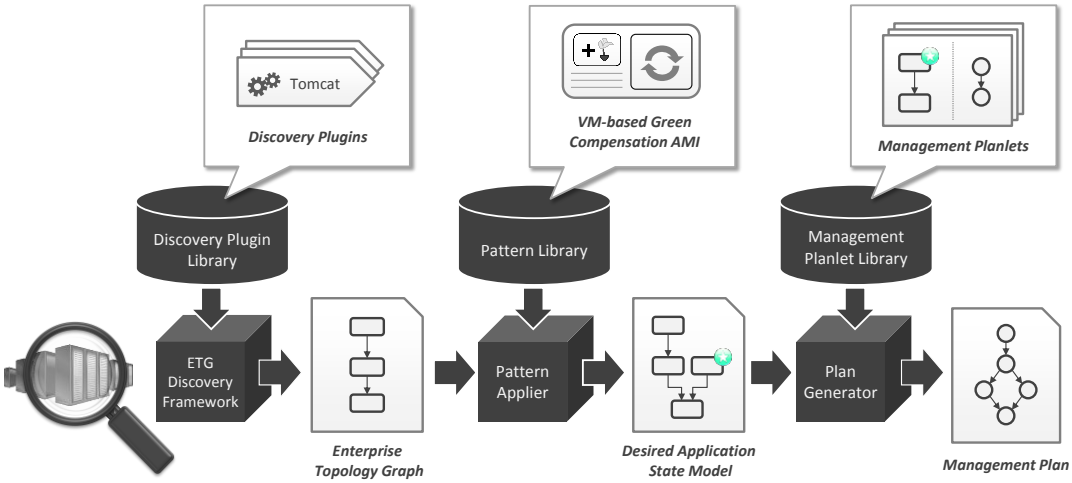

Figure 4. Overview of the Management Planlet Framework's pattern refinement and automation method (adapted from [9][12])

transformed fully automatically into executable workflows. In this section, we describe the management framework to provide required information. The framework's architecture and employed artifacts are shown in Figure 4.

We now describe how this framework can be used to apply a management pattern fully automatically to a running application. In the first step, the application's structure and all runtime information are captured as formal ETG. This model contains all nodes and relations of the process including runtime information in the form of properties, e.g., VM nodes provide IP-Address and SSH credentials. ETGs can be discovered fully automatically using the plugin-based ETG Discovery Framework [26]. Thus, this framework enables to automatically identify all VMs of a business process that are related to its environmental impact.

ETGs can then be transformed into *Desired Application State Models (DASMs)* through automatically applying management patterns. A DASM is a declarative description of management tasks to be performed on the application. It consists of (i) the application's ETG and (ii) one or more *Management Annotations*, which are declared on nodes and relations to specify management tasks to be executed on the associated element, e. g., that a given node shall be created, configured, or destroyed. Management Annotations define management tasks only declaratively, i.e., only *what* has to be done, but not *how*. Therefore, the framework employs a *Plan Generator* that consumes DASMs and generates executable workflows by orchestrating so called *Management Planlets*, which are small workflows that execute the Management Annotations declared on nodes and relations. Thus, they provide the *imperative* logic required to execute *declarative* Management Annotations in DASMs. Because the actual workflow generation is not important to understand the presented approach, we refer to Breitenbücher et al. [9][10][11][12] for more details and explanations.

To automatically transform ETGs into DASMs, the framework employs (i) *Semi-Automated Management Patterns (SAMPs)* and (ii) *Automated Management Idioms (AMIs)*, which both consume ETGs and output DASMs that describe the tasks to be performed. SAMPs implement the generic solution of high-level patterns and can be used to *semi-automate* the application of patterns. For example, the Green Compensation pattern is such a high-level pattern because it provides only generic information and requires additional manual refinement for its application. In contrast to this, an AMI provides a detailed refinement of a certain SAMP for a concrete use case. For example, the refinement of the Green Compensation pattern for Virtual Machines as shown in Figure 3 can be implemented as AMI that contains all required information to *fully automate* its application. Consequently, the output DASMs of SAMPs typically require additional manual refinement whereas the output DASMs of AMIs can be used directly to generate the workflows. Therefore, we realize the VM-based refinement of the Green Compensation pattern as fully automated AMI.

An AMI consists of two parts: (i) a *Topology Fragment* and (ii) a *Topology Transformation*. The fragment is a small segment of a topology used to check if the AMI is applicable to a certain ETG. Therefore, the fragment defines nodes and relations that must match elements in the ETG of the application in order to apply the pattern to the matching elements. The second part is a Topology Transformation that defines how to transform the input ETG to the DASM. Thus, it implements the AMI's refined solution by attaching Management Annotations to the affected nodes and relations. The resulting DASM then describes the management tasks to be performed on the application to apply the refined pattern.

D. VM-based Green Compensation AMI

In this section, we automate the refined Green Compensation pattern solution presented in Section X and show how the concept of AMIs can be used to implement a *VM-based Green Compensation AMI* that can be applied automatically to various VM-based business processes for compensating their CO_2 emissions. To create an AMI, the Topology Fragment must be defined first. In our scenario, this means that only a node of type *LinuxVM* must be defined as it (i) does not matter which underlying infrastructure is employed to host the VM, (ii) all Linux Virtual Machines enable remote access via SSH, and (iii) it does not matter

 ISBN: 978-1-61208-354-4

which components are hosted on the Virtual Machine. Thus, the AMI's Topology Fragment is as simple as shown in Figure 5 on the left and defines that the AMI can be applied to all nodes in ETGs that are of type *LinuxVM*.

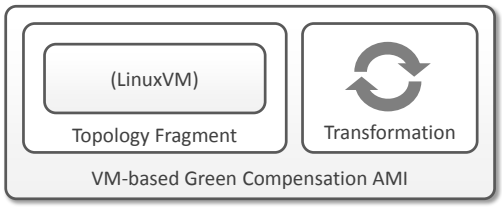

Figure 5. Automated Management Idiom (AMI) that refines the Green Compensation pattern for VM-based business processes

Secondly, the refined solution of the Green Compensation pattern shown in Figure 5 needs to be implemented as Topology Transformation. Therefore, the transformation has to modify the ETG for creating the DASM as follows: it has to (i) add a new node of type *Monitoring Agent* with attached *Create-Management Annotation* that defines that this node must be created, (ii) add a *hostedOn* relation from the Monitoring Agent node to the VM node including a Create-Management Annotation, (iii) add a CO_2 *Compensation Service* node without Management Annotation as this node already exists from third parties, and finally (iv) connect the Monitoring Agent to the Compensation Service node by a relation of type *uses* that must be created, too. Therefore, this relation has an attached Create-Annotation as well. To configure the Monitoring Agent node, an additional *Configure-Management Annotation* is attached to the Monitoring Agent node that defines that a configuration (also described in the Annotation) has to be set. This configuration can be implemented directly in the AMI.

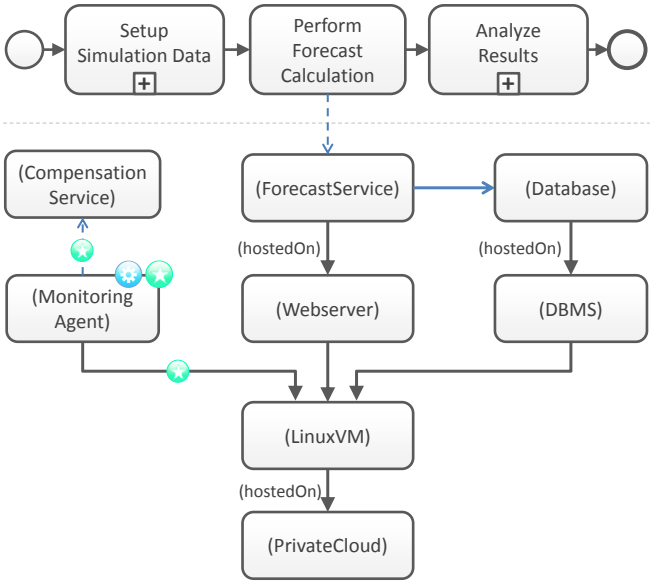

Figure 6. DASM resulting from applying the VM-based Green Compensation AMI to the motivating scenario

When this AMI is applied to a VM node contained in an ETG, the transformation will be applied and the corresponding DASM that can be used to generate the corresponding workflow is returned. Thus, the AMI is not bound to individual applications and can be applied generically to all business processes that contain Virtual Machine nodes in their ETG. Figure 6 shows the DASM that results from applying the described AMI to the motivating scenario. The green circles represent Create-Management Annotations while the blue circle represents the Configure-Management Annotation. The shown DASM defines exactly the management tasks to be executed for applying the pattern as described in the refined process shown in Figure 3. This DASM is then consumed by the framework's Plan Generator that searches and orchestrates appropriate Management Planlets implementing the required management logic to execute the management tasks described by the Management Annotations declared on nodes and relations in the DASM.

Consequently, this section showed how the requirements analyzed in Section II.A can be tackled for the automation of the Green Compensation pattern towards VM-based applications: first, the technical complexity is hidden completely by the AMI's transformation and the Management Planlets. Only the configuration of the monitoring agent may be up to the administrator, but this can be modeled also directly in the AMI's transformation if wanted. Secondly, the whole application of the pattern including its refinement for VM-based business processes is automated by the AMI. Thus, the result can be applied fully automatically to individual VMs to compensate their CO_2 emission. The implementation of our prototype proves the technical feasibility of the approach in general [9] [12].

However, the presented approach also requires significant effort that needs to be spent upfront. Automation is not free of charge but patterns need to be refined, AMIs need to be implemented as well as the corresponding Management Planlets. Whereas the definition of Topology Fragments is a rather trivial task the implementation of Topology Transformations may result in many unpredictable issues. The implementation of Management Planlets that capture these transformations require a lot of technical expertise regarding the employed technologies, e.g., connecting to a virtual machine using SSH and installing the required monitoring components is a non-trivial task due to firewalls and other security mechanisms. In addition, different structures of various kinds of VM-based applications must be considered in advance to provide a correct solution implementation. On the other hand the advantage is that these steps only have to be done once and the result can be used repeatedly. For example, Management Planlets can be used in various different automated management tasks and, therefore, the reusability aspect of this approach typically outvotes the effort that must be spent during the initial implementation. Moreover, the implementation of AMIs can be guided using a development method we presented in [27].

Besides the required effort and expertise the presented approach improves existing ones in the following aspects: (i) the generic Management Planlet Framework enables the

 ISBN: 978-1-61208-354-4

integration of various kinds of proprietary and heterogeneous technologies seamlessly into the overall Management Plan generation [10]. The refined solution that is implemented in our AMI is independent of individual technologies. Thus, it does not matter if the virtual machine is hosted on a public Cloud, for example Amazon EC2, or if a physical server is employed in the local infrastructure of an enterprise. (ii) the refinement of the Green Compensation pattern for use in VM-based applications demonstrates how non-functional requirements of the optimization of applications can be automated based on patterns. This automation is vital in the domain of Cloud application management. Therefore, the approach shows how the concept of patterns can be implemented to consider ecological aspects in enterprises without requiring the deep-technical knowledge that is mandatory to refine patterns manually each time. (iii) the automated refinement of patterns supports *economics of management*. The implemented transformation routines, i.e., the AMIs and Planlets, are reusable for individual applications and, therefore, further effort for managing these applications is minimized to a simple AMI selection.

The actual CO_2 emission of an application that can be compensated mainly depends on the individual application. Therefore, we are not able to present some dedicated figures representing absolute savings or improvements. Moreover, the approach shows holistically how improvements may be realized when using Cloud-based applications.

IV. Related Work

The Green Compensation pattern we used in this work is only one possible pattern to improve the CO_2 emissions of enterprises. In Nowak et al. [6][7], we present a bunch of patterns that can help to improve the negative environmental impact of enterprises. Moreover, more and more companies have own strategies to improve CO_2 emissions. Amazon AWS, for example, has built datacenters in Oregon that only use green hydro-electric power [28].

Falkenthal et al. [14] present an approach that enables capturing directly reusable implementations of patterns by linking them to the patterns they originate from. In addition, they describe how such *Solution Implementations* can be aggregated using explicit operators. Therefore, this approach could be used to link the workflow that implements the VM-based solution refinement directly with the Green Compensation pattern. However, this does not tackle the issue considering the technical complexity described in Section II.A: during the workflow's implementation, the technical complexity of accessing the VM, installing and configuring the agent, etc. must be implemented manually. This requires a lot of technical expertise that is shifted in our approach completely to the employed management framework: implementing the AMI's transformation requires no technical knowledge about the underlying technologies.

Fehling et al. [30] present an approach that enables annotating so called *Implementation Artifacts* to architectural patterns that can be used during their application to instantiate the pattern's solution. These artifacts may range from software artifacts and configuration files to executable processes that implement certain management functionality of concrete components that can be used to implement the pattern. The annotated artifacts may support and guide applying and refining patterns. However, the approach supports only manual pattern application. In our work, we focus on fully automated pattern application. In a further work, Fehling et al. [29] present how the application of *Cloud Application Management Patterns* can be supported by providing abstract processes that define the high-level steps to perform the solution. These abstract processes must be then refined *manually* for individual use cases. Despite these management patterns are interrelated with their architectural patterns that may provide reusable management flows in the form of Implementation Artifacts, the refinement approach remains mainly manual: the artifacts may be used to ease implementing parts of the solution process, but their orchestration must be done manually.

In Breitenbücher et al. [12], we showed how the *Stateless Component Swapping Pattern*, which is a pattern that enables migrating an application component without downtime, can be automated using the Management Planlet Framework similarly as presented in this paper. Thus, the suitability of using the framework for automating patterns is technically feasible in general. Therefore, the findings of Breitenbücher et al. [12] and this paper provide the basis for continuing the automation of patterns following this concept.

The original Management Planlet Framework [9] is extended in several publications. In Breitenbücher et al. [11], we showed how management tasks can be annotated with policies that define non-functional requirements on the execution of the task, e.g., security requirements such as the location of a Virtual Machine. In Breitenbücher et al. [10], we presented an approach to specify and integrate imperative logic into the declarative concept of DASMs. Thus, these extensions show that the Management Planlet Framework provides a flexible means for automating pattern.

V. Conclusion

In this paper, we addressed the gap between the abstract description of pattern solutions and their application to individual Cloud-based business processes. As the manual application of a pattern is typically very complex, time consuming, and error-prone we proposed an approach that is able to apply the Green Compensation pattern to individual use cases automatically over and over again. Therefore, we first described a method on how to refine the Green Compensation pattern to represent all necessary management tasks for its application. Next, we showed how the employed Management Planlet Framework is used to implement that concrete and detailed pattern solution in a reusable way.

In our future work, we (i) plan to extend that approach to support further Green Business Process Management patterns and (ii) evaluate the refinement of patterns from other domains to identify possible extensions.

Acknowlegments

This work was partially funded by the BMWi projects CloudCycle (01MD11023) and Migrate! (01ME11055).

REFERENCES

[1] A. Nowak, F. Leymann, D. Schumm, and B. Wetzstein, "An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering" ICT-GLOW 2011, Springer, 2011, pp. 150-164.

[2] A. Nowak, T. Binz, F. Leymann, and N. Urbach "Determining Power Consumption of Business Processes and their Activities to Enable Green Business Process Reengineering" EDOC 2013, IEEE, 2013, pp. 259-266.

[3] GreenBiz.com. How Will Sustainability Change at Your Company in 2011. http://www.greenbiz.com/blog/2010/12/29/how-will-sustainability-change-yourcompany-2011, retrieved 05.2014.

[4] Gartner Research. Gartner Identifies the Top 10 Strategic Technologies for 2010. http://www.gartner.com/it/page.jsp?id=1210613, retrieved 05.2014.

[5] Intl. Org. for Standardization. ISO14000. Environmental Management. http://www.iso.org/iso/iso14000, retrieved 05.2014.

[6] A. Nowak, F. Leymann, D. Schleicher, D. Schumm, and S. Wagner, "Green Business Process Patterns" PLoP 2011, ACM, 2011, in press.

[7] A. Nowak and F. Leymann, "Green Business Process Patterns - Part II" SOCA 2013, IEEE, 2013, pp. 168-173.

[8] D. Oppenheimer, A. Ganapathi, and D. Patterson, "Why do internet services fail, and what can be done about it?" USENIX 2003, ACM, 2003.

[9] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Pattern-based Runtime Management of Composite Cloud Applications" CLOSER 2013, SciTePress, 2013, pp. 475-482.

[10] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and J. Wettinger, "Integrated Cloud Application Provisioning: Interconnecting Service-Centric and Script-Centric Management Technologies" CoopIS 2013, Springer, 2013, pp. 130-148.

[11] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning of Cloud Applications" SECURWARE 2013, IARIA Xpert Publishing Services, 2013, pp. 86-95.

[12] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Automating Cloud Application Management Using Management Idioms" PATTERNS 2014, IARIA Xpert Publishing Services, 2014, pp. 60-69.

[13] C. Alexander, A Pattern Language. Towns, Buildings, Construction. Oxford University Press, New York, 1977.

[14] M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, and F. Leymann, "From Pattern Languages to Solution Implementations" PATTERNS 2014, Xpert, 2014, in press.

[15] OMG, Business Process Model and Notation (BPMN), Version 2.0.

[16] JET Tankstellen Deutschland GmbH. www.arktik.de, retrieved 05.2014.

[17] Deutsche Bahn AG. http://www.dbecoprogram.com /index.php?lang=en, retrieved 05.2014.

[18] MyClimate. http://de.myclimate.org, retrieved 05.2014.

[19] A. Nowak, F. Leymann, and D. Schumm, "The Differences and Commonalities between Green and Conventional Business Process Management" CGC 2011, IEEE, 2011, pp. 569 - 576.

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

[21] PrimaKlima. http://www.prima-klima-weltweit.de/co2/kompens -berechnen.php, retrieved 05.2014.

[22] EPA. http://www.epa.gov/cleanenergy/documents/egridzips/eGRID 2012V1_0_year09_SummaryTables.pdf, retrieved 05.2014.

[23] Sigar. http://www.hyperic.com/products/sigar, retrieved 05.2014.

[24] Migrate!. http://www.migrate-it2green.de/, retrieved 05.2014.

[25] CarbonFund. http://www.carbonfund.org/, retrieved 05.2014.

[26] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "Automated Discovery and Maintenance of Enterprise Topology Graphs" SOCA 2013, IEEE, 2013, pp. 126 - 134.

[27] U. Breitenbücher, T. Binz, and F. Leymann, "A Method to Automate Cloud Application Management Patterns", ADVCOMP 2014, IARIA Xpert Publishing Services, 2014, in press.

[28] R. Miller. Amazon's Cloud goes Modular in Oregon. http://www.datacenterknowledge.com/archives/2011/03/28/amazons-cloud-goes-modular-in-oregon/, retrieved 05.2014.

[29] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, "Pattern-Based Development and Management of Cloud Applications" Future Internet 2012, 4(1), pp. 110-141, 2012.

[30] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An Architectural Pattern Language of Cloud-based Applications" PLoP 2011, ACM, 2011, in press.

 ISBN: 978-1-61208-354-4