Pattern-driven Green Adaptation of Process-based Applications and their Runtime Infrastructure

Nowak, Alexander; Binz, Tobias; Fehling, Christoph; Kopp, Oliver; Leymann, Frank; Wagner, Sebastian

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

BibTeX:

```
@article {springerlink:10.1007/s00607-012-0188-x,
  author     = {Nowak, Alexander and Binz, Tobias and Fehling, Christoph and
                Kopp, Oliver and Leymann, Frank and Wagner, Sebastian},
  title      = {Pattern-driven green adaptation of process-based applications
                and their runtime infrastructure},
  year       = {2012},
  journal    = {Computing},
  pages      = {463-487},
  doi        = {10.1007/s00607-012-0188-x},
  volume     = {94(6)},
  publisher  = {Springer Wien}
}
```

Pattern-driven Green Adaptation of Process-based Applications and their Runtime Infrastructure

Alexander Nowak, Tobias Binz, Christoph Fehling,
Oliver Kopp, Frank Leymann, Sebastian Wagner

Institute of Architecture of Application Systems (IAAS)
University of Stuttgart, Universitaetsstr. 38, 70569 Stuttgart, Germany
Email: firstname.lastname@iaas.uni-stuttgart.de

Abstract: Business Processes are a key aspect of modern organization. In recent years, business process management and optimization has been applied to different cross-cutting concerns such as security, compliance, or Green IT, for example. Based on the ecological characteristics of a business process, proper environmentally sustainable adaptation strategies can be chosen to improve the total environmental impact of the business process. We use ecological sustainable adaptation strategies that are described as Green Business Process Patterns. The application of such a Green Business Process Pattern, however, affects the business process layer, the application component and the infrastructure layer. This implies that changes in the application infrastructure also need to be considered. Hence, we use best practices of cloud application architectures which are described as Cloud Patterns. To guide developers through the adaptation process we propose a pattern-based approach in this work. We correlate Cloud Patterns relevant for sustainable business processes to Green Business Process Patterns and organize them within a classification. To provide concrete implementation support we further annotate these Cloud Patterns to application component models that are described with the Topology and Orchestration Specification for Cloud Applications (TOSCA). Using these annotations, we describe a method that provides the means to optimize business processes based on Green Business Process Patterns through adapting the implementation of application components with concrete TOSCA implementation models.

Keywords: Green Business Process Pattern, Cloud Pattern, Green IT, TOSCA, Adaptation of Applications, Ecological Sustainable Business Processes

Acknowledgements: This work was partially funded by BMWi project Migrate! (01ME11055) and the BMWi project CloudCycle (01MD11023). Further we want to thank our colleagues at the IAAS for the fruitful discussions.

1 Introduction

Business processes are one of the key assets of organizations. They define explicitly which input is performed in which way by which participant for what purpose and what the output is [1]. In order to be a successful market participant business processes need to be adapted to changing environments continuously. Business Process Management (BPM) provides one approach to tackle this issue [2]. By applying BPM, organizations try to realize cost reductions, quality improvements, time savings, and increased flexibility [3]. However, more and more cross-cutting concerns arise and require the extension of current BPM concepts and methods. Besides cross-cutting concerns such as security and compliance, one challenge is the proper application of BPM in the context of ecological sustainability and Green IT [4][5][6].

Existing technologies allow the annotation of business processes with different indicators and information [6][7][8]. Based on this information, an adaptation of the business process as well as the underlying services and infrastructure is desired. This often requires the application of best practices from the domains of business process re-engineering, application architecture design, and infrastructure optimization. As a consequence, to make a process-based application more eco-efficient all of these aspects need to be considered. In our current research with industry partners and within the German government founded project Migrate! [9], we identified best practices applied on these different application aspects and organized them in a pattern catalog: We described different Cloud patterns in [10] and [11] and green business process patterns in [12]. Details on sustainable business processes and sustainable process metrics can be found in [6] and [14]. Thus, we omit a detailed description in this paper.

To effectively realize these patterns we propose to annotate the green business process-centric patterns with patterns from other domains, especially the Cloud patterns [11], to determine necessary changes on the different application aspects following common ecological goals. Consequently, the contribution of this work is twofold: We first propose a classification describing the relationships between the different patterns. Additionally, we show their relation to concrete implementation artifacts described within a specific service description specification. Second, we show how the annotation of runtime-specific application models to patterns guides the adaptation of an existing application towards a better ecological footprint. We propose a novel method how these relations can be utilized in order to realize concrete implementations for green business process patterns.

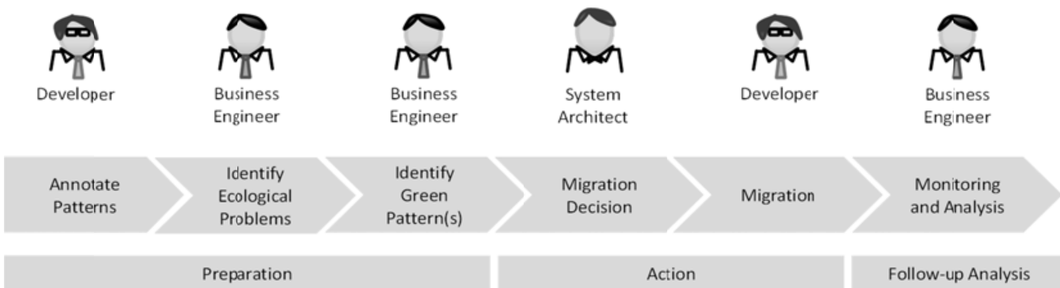

Fig. 1 Pattern-driven adaptation process

Fig. 1 describes the proposed approach as a process including the corresponding stakeholders. These stakeholders are derived from [2]. The approach is divided into three different phases: the *preparation* phase comprises all steps necessary to apply the proposed method of this work. The method to adapt business processes and their underlying infrastructure is represented by the *action* phase. After that phase, the results of the performed adaptations need to be evaluated in the *follow-up analysis* phase. In this work, we mainly present the method used in the action phase.

With respect to the complete approach shown in Fig. 1, we assume that (1) an existing business process is already implemented and used within a certain business environment, and (2) the corresponding applications or application components, respectively, supporting this business process are already in place. In the first step of the approach, a *developer* needs to annotate the green business process and Cloud patterns to the organization's available application components implementing these patterns. Therefore, the application components are described with the Topology and Orchestration Specification for Cloud Applications (TOSCA) [13]. Next, a *business engineer* identifies the ecologically critical parts of the running business process. Based on those findings the *business engineer* may select one or more adequate green business process patterns that fit into the general strategic objectives of the organization. The green business process patterns describe how common best practices can be utilized in order to adapt a business process towards a more eco-efficient one in an abstract manner. The concrete adaption of the process implementation is then made by the *system architect*. Based on existing Cloud patterns that are annotated to the green business process patterns the system architect may decide which Cloud patterns are able to realize the chosen green business process pattern(s) best. Subsequently, the patterns for concrete implementation are applied by the *developer*.

In this work we provide a solution that abstracts from the concrete use-cases of our industry projects and partners. However, we use the existing problem statements from sustainability concerns in IT systems to show the feasibility of our approach. Other topics of interest may be put in place. The remainder of this paper is structured as follows: Section 2 provides fundamentals used in this work mainly focusing on patterns in general as well as TOSCA. The concrete patterns and their relations to each other are described in Section 3. For the determination of a concrete implementation solution Section 4 describes how patterns can be annotated with TOSCA Service Templates. Subsequently, Section 5 provides an end-to-end example describing the method utilizing patterns and TOSCA Service Templates in order to provide concrete adaptation strategies for process-based applications.

2 Fundamentals

In this section we provide the fundamentals required for the subsequent sections. Section 2.1 presents a short overview on Cloud computing and research results. In Section 2.2, we provide a summary on patterns in the context of Cloud and sustainable computing. Section 2.3 provides an overview on TOSCA.

2.1 Cloud Computing

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources […]" [27]. Cloud computing is mainly applied to turn capital expenditure (CAPEX) into operational costs (OPEX) [28]. The underlying technologies and concepts are: (i) virtualization, (ii) elasticity, and (iii) utility computing [29]. An application may be built from scratch to use the benefits of Cloud computing. However, it is more common to move existing applications to the Cloud. Leymann et al. [23] propose a generic framework to find the optimal splitting of an application onto different clouds, based on labels attached to the application's components. Trummer et al. [25] optimizes which parts of an application are hosted on which cloud. Binz et al. [24] proposes a technical realization to migrate applications which use adapters to host application components on previously incompatible Cloud environments. Furthermore, a Cloud provider itself has to find an optimal way to utilize his resources. One method is to share existing application instances among different tenants. In that context, Fehling et al. [26] show how a provider may optimally distribute tenants in applications.

In this paper, we also move applications to the Cloud. In contrast to the other works, we use green business process patterns and Cloud patterns to *change* the existing application environment and *move* applications or parts thereof to the Cloud.

2.2 Patterns

Process-based applications usually consist out of loosely coupled application components. These components are often developed for specific service providers. To describe the requirements of applications in a more generic way we have abstracted from concrete developing guidelines [11][12]. We use patterns as a fundamental aspect of this approach. They guide developers in creating and adjusting application's architecture, infrastructure, and the business process on top of it. In general, patterns represent common solutions for common and reoccurring problems in an abstract form, i.e., they abstract from concrete implementations. We argue that an abstraction of solutions to realize design decisions for process-based applications is beneficial for reusing solutions in cross-cutting domains. Consequently, the application of patterns allows specific implementations for different service providers without defining specific implementation details within these patterns. In this work, we are using in particular two types of patterns: *Cloud patterns* (described in Section 3.1) and *green business patterns* (described in Section 3.2).

2.3 TOSCA

The *Topology and Orchestration Specification for Cloud Applications* (TOSCA) [13] is currently developed in a newly established OASIS Technical Committee [16]. The goal of TOSCA is to improve the portability and manageability of applications, including infrastructure, software, and their management. TOSCA provides a format for declaratively describing a service's application topology (so-called *Topology Template*) and a description how to manage this application during the applications lifecycle (represented by so-called *Plans*), for example, how to deploy or terminate a service.

Such a package is called a *TOSCA Service Template*. With this, TOSCA specifies a generic language to model services and their management. However, a TOSCA Service Template does not provide a representation of a business process that is using the application components. That is, the relations between the corresponding activities of a business process are not captured but the application topology underneath.

In this paper we use TOSCA Service Templates as means for describing the artifacts of a process-based application. We do not extend TOSCA but merely show its usage in application adaptation. In the following, we present an overview on the concepts of TOSCA to support better understanding of its capabilities.

The topology is a graph of nodes (*Node Templates*) and edges (*Relationship Templates*), which represents the structure of the service. Both Node and Relationship Templates reference a type, the *Node Type* and *Relationship Type* respectively. The type defines the kind of properties and the lifecycle the templates must follow. The Node Type, for instance, references an XML Schema Definition (XSD) [17] for properties (e.g., IP address) and defines the states of the lifecycle of this specific type (e.g., stopped, starting, started, and paused), which both must be followed by the Node Template. Nodes (Node Types and Node Templates) specify how they are realized using *Implementation Artifacts,* which in turn may reference artifacts of arbitrary type. It is possible to reference multiple artifacts in one node and let the middleware which deploys this cloud service decide which type of artifact to choose. For example, a Linux Node Template may have multiple virtual machine images referenced as Implementation Artifacts, one for each type of supervisor the virtual machine should run on. This is one concept which helps modelers to define portable cloud services which can be operated on different clouds. Another important concept for Node Types is the definition of *Interfaces* which define the operations possible on Node Templates of this type. Interfaces define which management operations are supported by this node, for example, to startup, deploy, or update the entities represented by the node. The nodes are connected by relationships which are binary (i.e., connect two nodes), directed, and may have cycles.

Plans are used to manage the lifecycle of an application. They can be provided in an arbitrary workflow language, although BPMN [15] is the preferred one. The idea driving this is that the organization which has the experience and knowledge how to operate and manage this cloud service models their knowledge and best practices as plans. This enables the management of cloud services to be reusable and portable, because users are able to execute predefined plans without requiring extensive knowledge of the service itself. A wide range of service management aspects can be captured in Plans: setting up a service, adding a user, or updating software, for instance. Plans orchestrate the interfaces defined on nodes into higher level functionality spanning multiple nodes. By using standard workflow languages arbitrary external Web services can be invoked as needed by the respective plan, for example, to acquire licenses or get approval. The usage of these plans and the topology in the context of adaption of process-based applications is detailed in Section 4.

3 A Consolidated Pattern Catalog

For the basis of the consolidated pattern catalog we use different Cloud and green business process patterns. An overview of the used patterns is presented in Section 3.1 and Section 3.2, respectively. More details on each pattern can be found in [11] and [12]. In Section 3.3, we show how these patterns are related to each other by identifying a correlation matrix.

3.1 Cloud Patterns

We consider process-based applications managed in cloud environments to be comprised of multiple loosely-coupled application components hosted on different sites. In the scope of green business process reengineering these application components may be migrated between different cloud environments or between a static data center and the cloud to achieve an ecological improvement. However, the industry-driven evolution of cloud offerings often obfuscates the underlying concepts and architectural principles. This makes the functionality offered by cloud providers hard to compare and architectural changes in the application, in order to move between different cloud environments, are difficult to determine.

The challenge of making cloud providers interchangeable is targeted by TOSCA which provides a common language to describe application topologies. Developed application components make assumptions on each other and their runtime environment. These assumptions are not covered by a TOSCA description. However, they also have to be considered when moving components between environments. For instance, the availability of virtual machines found in public clouds is often lower than the availability guaranteed by private cloud environments. During the migration between such different environments, these different properties have to be regarded by changes in the application architecture to cope with component failure, for example. In [10] we propose a set of patterns describing the architectural principles of different clouds and their offerings as well how to build applications on top of such offerings. An overview of the different patterns and their classes is shown in Fig. 2. We further extended the use of the pattern format to additionally describe (1) how clouds offer resources (by stating the cloud service model), (2) who is accessing the clouds (by stating the cloud type), and (3) how the services of the offerings found in these clouds behave (by stating the cloud offering). This allows the classification of clouds and their offerings with respect to the supported patterns and enables the correlation of architectural patterns that application developers need to follow when using a concrete cloud offering.

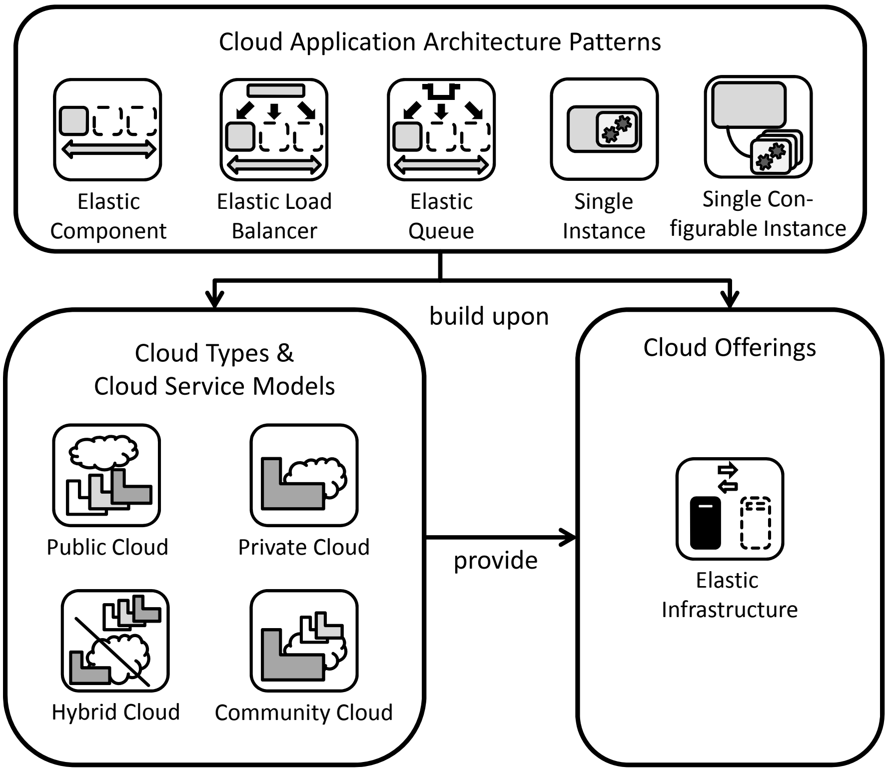

Fig. 2 Different classes of cloud patterns

In green business process reengineering, one important aspect is where the IT resources are hosted. Therefore, patterns describing the different cloud environments such as *private cloud*, *public cloud*, *community cloud*, and *hybrid cloud* are relevant. Second, elasticity of the application may be used to enable a more ecological use of cloud resources, aside the cost reduction to which it is mainly associated otherwise. In the application architecture, this elasticity is often enabled by an *elastic infrastructure* in which virtual services may be started and stopped on demand. In addition, the application components have to support elasticity by being removable from the application, if they are not needed. The *elastic component* is scaled based on infrastructure utilization, such as CPU or memory, while the *elastic load balancer* and the *elastic queue* consider requests send to the scaled component in a synchronous and asynchronous fashion, respectively. Another way to make applications more ecological friendly is sharing resources. The architectural patterns to enable this resource sharing, so called multi-tenancy, are therefore also relevant for this approach: A *single instance* is a component that is accessed by different tenants. If these tenants require a custom configuration, a *single configurable instance* may be used which accesses individual configuration files for each tenant to adjust its behavior [18].

The presented Cloud application architecture patterns are closely related to ecological concerns. If these patterns are used, other architectural patterns may be referenced. The additional patterns provide a more detailed description of architectural changes in the application. For example, consider an application running in a private cloud. This environment is described by the private cloud pattern. Now, the system architect decides to move parts of the application into a public cloud. Therefore, the environment changes from being a private cloud to a hybrid cloud. In this new environment, the system architect is faced with the challenge to integrate the two cloud environments to make application components accessible from each other. This may, for example, be hindered by firewall restrictions. To guide the system architect during this migration process the cloud component gateway pattern is recommended by [10] as architectural change to the application.

3.2 Green Business Process Patterns

Organizations already spend notable effort in becoming more sustainable. The drivers for this trend are manifold: (1) increasing costs for energy and raw-materials, (2) legislative regulations to reduce the emission of carbon dioxide, or (3) the public appearance of the organization. Therefore, we observed reoccurring forms of sustainable business process design in offerings of several companies and in recent work on Green IT and Green IS. In addition, we also observed how existing forms of process (re-)engineering are applicable in the context of improving the ecological impact of business processes. We have identified and explicated these existing forms of sustainable business process design as *green business process patterns* [12]. Consequently, the green business process patterns provide solutions to reoccurring problems in process design with respect to the corresponding environmental impact of a process. There are two different classes of patterns we have identified:

Basic patterns extend or change a process by providing explicit sustainable alternative services, products, or resources. We identified the following patterns: (1) *Green Compensation* introduces a compensating activity or process whenever the original process structure and resources may or will not be changed. (2) *Green Variant* provides alternative sustainable activities or processes while the original process remains active. (3) *Resource Change* describes the exchange of resources (of any kind) in use. (4) Green Feature addresses the change of a service replacing a certain part with a more sustainable alternative that can be promoted in public relation.

Process-centric patterns achieve a more sustainable process design by directly modifying the structure or the way a process is performed. We identified the following patterns: (1) *Common Process Improvement* puts common process optimization in the context of green process design. (2) *Process Automation* is meant to decrease the environmental impact by automating processes, and (3) *Human Process Performance* describes the other way round – performing processes by humans. (4) *Insourcing* and (5) *Outsourcing* addresses the aspects on whose site a specific process or parts thereof should be performed.

An overview of the identified patterns is presented in Fig. 3.

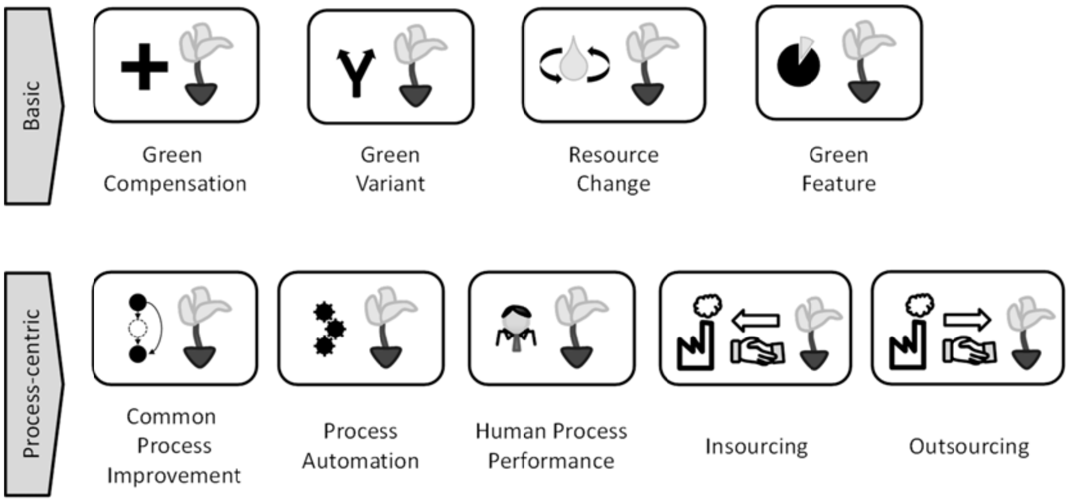

Fig. 3 Overview of green business process patterns

3.3 Correlation and Classification of Patterns

A correlation of the different patterns is required to guide developers during the identification of how to transform the adjustments proposed by the green process patterns in actions applied to the application's architecture and infrastructure. An overview of the correlation between green business process patterns and Cloud patterns is presented in Fig. 4. Based on the proposed pattern catalog we have identified three different layers relevant for adapting applications: (1) the *business process layer,* which is focused on changes to the business process model. (2) The *application architecture layer*, which is focused on the design and implementation of application components and services used by the process. (3) The *infrastructure layer*, which deals with the resources used for providing the application runtime.

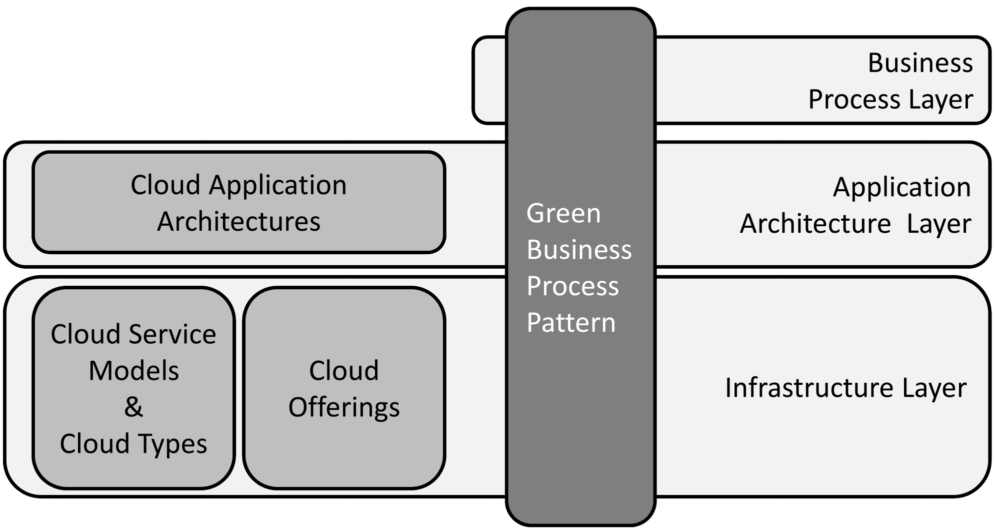

Fig. 4 Correlation of used patterns

The proposed patterns are now classified within the three layers that correlate to the three aspects of applied best practices. Different types of Cloud patterns address different layers. *Cloud offerings*, *Cloud service models*, and *Cloud types* can be assigned to the *infrastructure layer* as they describe the infrastructure resources used for a specific application. The *Cloud application architecture patterns* can be assigned to the *application architecture layer* as they describe how applications components are build. The *business process layer* is currently not covered explicitly by the proposed cloud patterns. However, process-based applications are often developed out of loosely coupled components. Thus, changes in the *application architecture layer* may also influence the *business process layer*. The result of the classification is shown in Fig. 4.

In contrast to the cloud patterns, *green business process patterns* are defined in a more abstract way. As a result different classification layers can be affected within a single pattern. The *Resource Change* pattern, for example, may be applied either on the *infrastructure layer*, e.g., by exchanging a server component, the *application architecture layer*, e.g., by exchanging an applications component, or both. Additionally, the business process may be changed as well in order to support the new resources within the business process adequately. Besides the cross-references within the pattern catalog a precise correlation between green business process patterns and Cloud patterns is necessary for a concrete adaptation of process-based applications. This correlation helps to actively guide an application developer on how to (1) identify possible implementation solutions based on the Cloud patterns and (2) identify other Cloud patterns from the catalog necessary for adaptation, e.g., patterns useful or required when moving components from private to public cloud environments.

Table 1 Cloud and green business process pattern relation matrix

	Green Process Pattern / Cloud Pattern	Green Compensation	Green Variant	Resource Change	Green Feature	Common Process Improvement	Process Automation	Human Process Performance	Outsourcing	Insourcing
Cloud Environments	Public Cloud	X		X		X	X		X	
	Private Cloud	X		X	X	X	X			X
	Hybrid Cloud	X	X	X		X	X		X	
	Community Cloud	X		X		X	X		X	
Elasticity	Elastic Component		X				X		X	X
	Elastic Load Balancer		X				X		X	X
	Elastic Queue		X				X		X	X
	Elastic Infrastructure		X	X			X		X	X
Multi-Tenancy	Single Instance		X		X		X		X	X
	Single Configurable Instance		X		X		X		X	X

The correlation matrix presented in Table 1 describes pattern relations relevant for implementing a green business process pattern. The matrix describes which cloud pattern support the intent of a green business process pattern. Correlations are marked by an “X”. As stated in Section 3.1, the Cloud application architecture patterns can be divided into three different categories: *cloud environments*, *elasticity*, and *multi-tenancy*. These categories are shown in the left side of Table 1, represented by the rows of the table. The green business process patterns are represented by the columns of the table.

In the following we provide details of the correlations shown in Table 1:

(1) *Green Compensation* introduces additional services to existing infrastructures in order to compensate parts of their environmental impact. Therefore, different *Cloud environments* are relevant to provide that services.

(2) A *Green Variant* directly affects the process model and the underlying application components. Therefore, an important green aspect is elasticity of application components. Elasticity patterns enable an efficient usage of resources by flexibly provisioning and de-provisioning application components on demand. Additionally, creating a new variant usually includes different environments, subsumed as *hybrid Cloud*.

(3) The application of the *Resource Change* pattern usually affects the infrastructure used by an application and does not affect the application components themselves. Thus, any cloud offering that provides a sufficient elastic infrastructure is able to decrease the total energy consumption. An elastic infrastructure allows demand-specific usage of resources, i.e., whenever a resource is not needed it can be turned off.

(4) *Green Features* that are related to a process or to a product are usually performed within the corresponding organization. Thus, this pattern is correlated to the *private Cloud* pattern. Furthermore, a Green Feature can be achieved by changing the usage of a cloud application. Using multi-tenancy functionality, for example, provides a way more efficient provisioning of an application than regular provision techniques, where each customer is served with an own application and hardware instance.

(5) *Common Process Improvement* basically addresses the structure of business processes and the resources used to perform it. Changes in this structure or exchanging the underlying resources of a process and its activities requires to set-up new *Cloud environments*.

(6) *Process Automation* often provides the possibility to create new application components and to individually define the used infrastructure. Therefore, all Cloud patterns from Table 1 provide suitable solutions: (a) put application components to cloud environments in order to save computing resources, (b) build *elastic* application components to switch them on and off on demand, or (c) use *multi-tenancy* functions to save resources.

(7) *Human Process Performance* is not applicable to our approach as this pattern describes solutions for human-centric business process activities. Our approach, however, focuses on automatic business processes in IT environments.

(8) *Outsourcing* may affect all Cloud Patterns except the *private Cloud*: a private cloud is operated within the property of the organization running the corresponding business process. All other cloud patterns can be applied to improve resource usage.

(9) *Insourcing* aims at running services within an organizations own property. Thus, this pattern is correlated to the *Private Cloud* pattern. However, advantages provided by *elasticity* and *multi-tenancy* functionality in cloud environments can also be achieved in a *private Cloud* environment.

4 TOSCA Annotation of Pattern Catalog

In the last section, we described the correlations of the different patterns that help developers to identify proper adaptation strategies for process-based applications. The proposed green business process patterns as well as the Cloud patterns, however, only describe an implementation independent solution for a problem in a generic way. There is still a lack of how to provide concrete implementation guidance for process-based applications based on the patterns in use. We propose to use TOSCA (see Section 2.3) as one suitable solution to address this problem.

The approach of this work provides a new method for adapting process-based applications using patterns. We postulate the following: (1) The application to be adapted is already described in TOSCA format and (2) the applications or components thereof used for adapting the application of interest are also described in TOSCA format. This implies the existence of several independent TOSCA Service Templates: The Service Template of the existing application that should be adapted and other TOSCA Service Templates describing self-contained services that are used for adapting the existing applications TOSCA Service Template, i.e., replacing part thereof.

To utilize TOSCA Service Templates for concrete adaptation strategies, we first annotate the corresponding templates to both green business process patterns and Cloud patterns (Fig. 5). The annotation of Cloud patterns by TOSCA Service Templates enables developers to identify possible pattern implementations. In some cases, such as the application of the *Green Compensation* pattern, it is also necessary to annotate a TOSCA Service Template directly to a green business process pattern. For instance, a new service may be introduced that provides a donation mechanism for climate projects. There is no need to change the current application architecture. The business process implemented by the application, however, needs to be changed to support the introduced donation activity. This change is not part of the TOSCA Service Template.

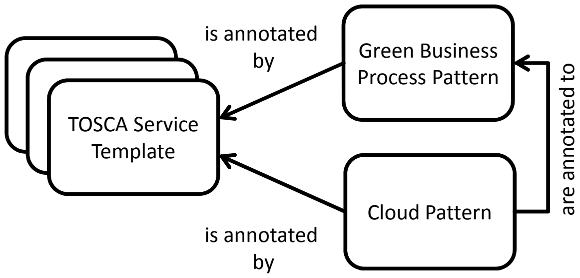

Fig. 5 Relations between TOSCA Service Templates and used patterns

To adapt a given application described by a TOSCA Service Template, we need additional information on how the topology of the given applications actually looks like, i.e., which components are used in which way. The application topology of an application described in TOSCA contains one or more Node Templates that describe the components a service is built of. To find proper substitutes for adapting either the whole application or specific parts thereof these Node Templates need to be annotated by the proposed Cloud patterns that indicate which patterns are used for the implementation of this node. Fig. 6 shows the corresponding extension of Fig. 5. Note that the proper matching of annotated Cloud patterns to our pattern catalog is vital in order to identify suitable adaptation alternatives including their actual implementation. As described in Sect. 3.3, the Cloud pattern catalog provides also information that help developers to identify required Cloud patterns that should be used when performing a certain adaptation, e.g., useful patterns that should be implemented when moving an application component from a private to a public Cloud.

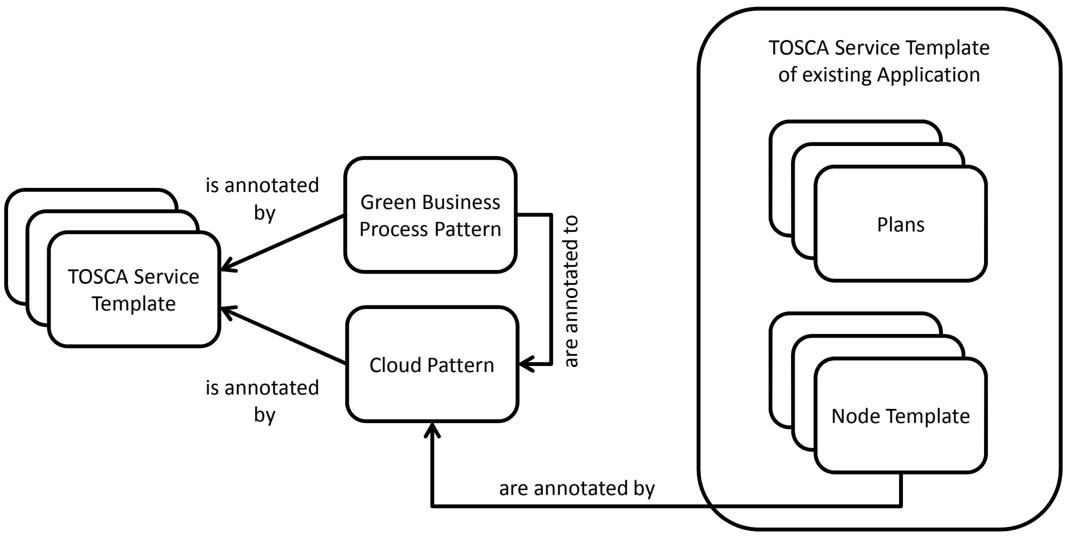

Fig. 6 Annotation of TOSCA Node Templates

Fig. 6 presents the required annotations for applying green business process patterns. If the environment is set-up like shown in Fig. 6, an application is ready for adaptation and a corresponding migration strategy can be identified. The abstract process of our approach is presented in Fig. 1. Appropriate stakeholders identify green business process patterns they want to apply based on their organizations strategic objectives. This first step determines in which way a process should be adapted. Proper adaptation alternatives that fulfill the required functional and non-functional requirements of the application can be identified using the relation of green business process patterns to the Cloud patterns catalog and the annotation of the Cloud patterns to the underlying architecture components. The result is a set of TOSCA Service Templates that qualify as adequate adaptation alternatives. Again, the resulting set of TOSCA templates may be annotated to a green business process pattern or to a Cloud pattern. The concrete annotation depends on the effects the adaptation of the application has to the layers introduced in Fig. 4. Consequently, by adapting the TOSCA Service Templates that are associated to an application via the Cloud patterns, the changes proposed by green business process patterns can be realized. To realize an adaptation strategy successfully, we need to extend our postulations at this point: (3) the annotations of a node within the topology template of a TOSCA Template need to be in the same format as used in the cloud pattern catalog. This enables proper matchmaking between the different the patterns annotated and the ones from the catalog. (4) Due to exchangeability and reusability reasons we only use self-contained TOSCA Service Templates for each part that should be adapted within a TOSCA Service Template. That means each Node Template to be exchanged is replaced by another TOSCA Service Template that is imported into the existing one.

Based on our assumptions the adaptation of an application (or parts thereof) always requires adapting the original TOSCA Template that describes the complete application of interest. These changes may affect both the Topology Template as well as the Plans that are included. Within the Topology Template single Node Templates or groups of Node Templates may be replaced by a new Node Template that implements the new functionality. Changing parts of a TOSCA Service Template affects both the relations between different Node Templates and the dependencies to existing Plans. To exchange specific Node Templates TOSCA provides import functionality for other TOSCA Templates that allows the usage of Node Templates thereof. In some cases it might also be sufficient to only adapt existing or add new Plans to a given TOSCA Template. For example, a new Plan may describe how a specific component is suspended in times of no usage. The Topology Template is not affected in this case.

5 Method Usage

To illustrate the method, this section presents an end-to-end example scenario for pattern-based business process reengineering. Section 5.1 provides an overview of the scenario. Sections 5.2 to 5.7 describe the application of the proposed method within this scenario as depicted in Fig. 1. The presentation focuses on the performance of the proposed method. Related methods on how to gather the necessary input information (such as energy consumption or CPU usage) are out of scope of this work.

5.1 Overview of the Scenario

Fig. 7 depicts a sample BPMN business process, which is performed by an architecture office to plan and implement a bridge construction. The process is triggered when the architecture office receives a bridge construction order from a customer. This order is created and managed in their customer relationship management system (CRM). Based on the customer requirements such as bridge length, carrying capacity of the bridge or the geographic structure, the CAD system is used to develop a new computer model of the bridge. For storing CAD models and for managing CAD model templates a corresponding repository is attached to the CAD system. After a first draft of the CAD model is created, several simulations are performed on the model. For instance, one simulation verifies the stability of the bridge under extreme conditions such as hurricanes or earthquakes. As these kinds of simulations are very resource intensive, the simulation environment is hosted on high performance compute nodes. In case one simulation reveals shortcomings in the bridge model, it has to be optimized and the simulations have to be executed again with the optimized CAD model. The CAD model optimization and simulation steps are performed until the simulation results ensure that the bridge meets all the requirements imposed on it. In the last step of the process the architecture office commissions a construction company to build the bridge based on the developed CAD model.

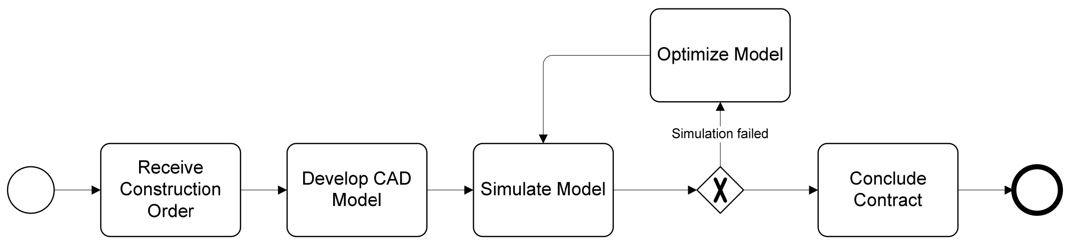

Fig. 7 Construction Process of a bridge

5.2 Annotate Patterns with TOSCA Service Templates

To identify proper adaptation scenarios the developer first needs to annotate available TOSCA Service Templates to the patterns from the pattern catalog described in Section 3.3. He may either annotate own or third party TOSCA Service Templates. Depending on the patterns used the developer may also define new TOSCA Service Templates that represent specific functionality required for a specific infrastructure. The TOSCA Service Templates annotated to the patterns define the result set of possible adaptation actions used for the migration (see Section 5.5 and Section 5.6).

5.3 Identify Ecological Problems

To improve the environmental impact the architecture office decides to reduce their ecological footprint. The corresponding stakeholders decide to use the green business process re-engineering approach as described in [6] and they use green process patterns to improve the ecological footprint of their processes. This task encompasses the identification of environmentally harmful elements in the process model. The sustainability of the process model is measured by monitoring the CO_2 emission of the process instances, which is derived from the corresponding energy consumption of the underlying infrastructure. The main infrastructure components used by the process are shown in Fig. 8.

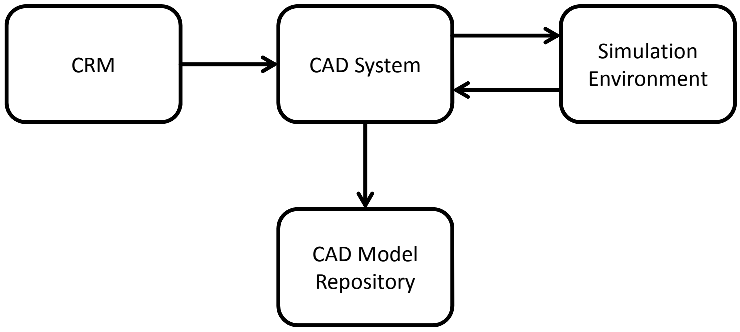

Fig. 8 Application components used for executing the business process

To obtain energy information, sensors are added to the hardware components where the process activities are executed on (e.g., the CRM software or the simulation environment). Using these measurements the business engineer observes a high CO_2 emission when the activity *simulate model* is performed. The reason for that are high performance servers that are required by the simulation environment. The impact on the total CO_2 emission of the process is even more critical since the *simulate model* activity is usually performed several times during the execution of a process instance. Moreover, the business engineer detects that these servers waste a lot of energy even when the *simulate model* activity is not executed, i.e., when the servers are idle. More details on how to analyze business processes with respect to their environmental impact can be found in [6] and [19].

5.4 Identify Green Pattern(s)

As described in Section 4, the annotations of TOSCA Service Templates allow proper matchmaking of patterns in use and patterns from the pattern catalog. Based on the identified process activities being environmentally inefficient, the business engineer suggests the architecture office to apply two green business process patterns to their business process: The *Resource Change* pattern and the *Green Compensation* pattern, which they decide to apply. The *Resource Change* pattern is used to improve the energy consumption of the *simulation environment*. Furthermore, the *Green Compensation* pattern is used to compensate parts of the energy consumption of the CAD System. Utilizing the Cloud pattern catalog proper adaptation patterns can be identified as described in Section 3.3. The annotated TOSCA Service Templates then provide concrete implementation support. In the following we will explain how the two green business process patterns are realized by applying the proposed method.

5.5 Migration Decision

After deciding which green business process patterns should be used to reduce the CO_2 emission of the business process and the underlying infrastructure, the architecture office applies the further steps of the method presented in this paper. The architecture office already created a TOSCA Service Template of their application as described in Section 2.3.

Fig. 9 shows how the application components described in Fig. 8 are depicted as TOSCA Service Template. Due to the fact that TOSCA does not provide a graphical representation yet we decided to use a common representation of application components. The different elements are used as follows: *Node Templates* are depicted as rectangles with a label referring to the name of the node. *Relationships* between different *Node Templates* are depicted as arcs with a label referring to the semantics of the relationship. Annotations are depicted as dashed lines and a label referring to the corresponding pattern. According to the TOSCA specification each *Node Template* refers to a *Node Type* that describes the implementation of a *Node Template*. For better readability *Node Types* are not depicted in Fig. 9. Furthermore, the TOSCA Service Template also includes the *CRM* system and the *cad model repository*. These components are internally provided as services. They, however, are not affected by the business process optimization. Thus, we do not show the complete application stack of these components in Fig. 9. For the components *CAD software* and the *simulation software* that are considered for adaptation the whole application stack is shown.

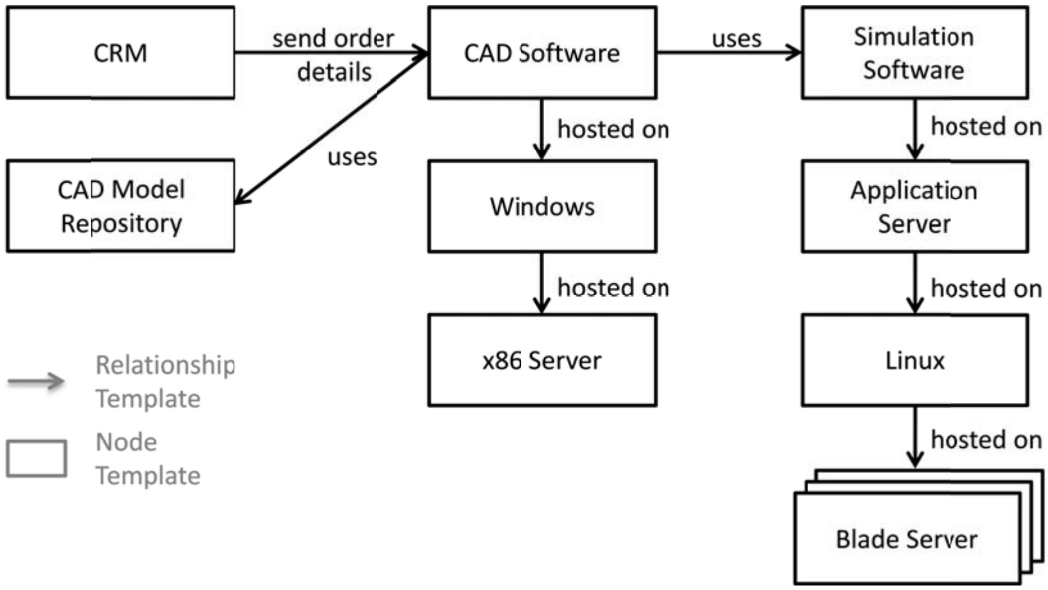

Fig. 9 Applications used for process execution described as TOSCA Service Template

To find proper alternatives for the application components in use the architecture office needs to annotate Node Templates of their TOSCA Service Template with applicable Cloud patterns presented in Section 3.1. Referring to Section 3.1 and Section 3.3 we assume that the TOSCA Service Template is annotated with Cloud patterns from the classes depicted in Fig. 2, for example *Cloud Type* (e.g., Private Cloud) and *Cloud Service Models* (e.g., Infrastructure as a Service). The system architect and the developer have made the following annotation:

(1) The *blade server* with a *linux* system on top is annotated with the *Infrastructure as a Service* pattern. This part of the application stack corresponds to Infrastructure as a Service offerings provided by Cloud providers. Additionally, this part is also annotated as *Low-available Computing Node*. This pattern refers to the availability requirements of the compute node.

(2) The *blade server* with a *linux* and an *application server* on top is annotated as *Platform as a Service* pattern. This part of the application stack corresponds to Platform as a Service offerings provided by Cloud providers.

(3) The *CRM* and *CAD model repository* are both annotated with the *Software as a Service* pattern. This reflects the service-oriented usage of these components.

(4) Additionally, all Node Templates are annotated with the *Private Cloud* pattern.

The annotated TOSCA Service Template is shown in Fig. 10.

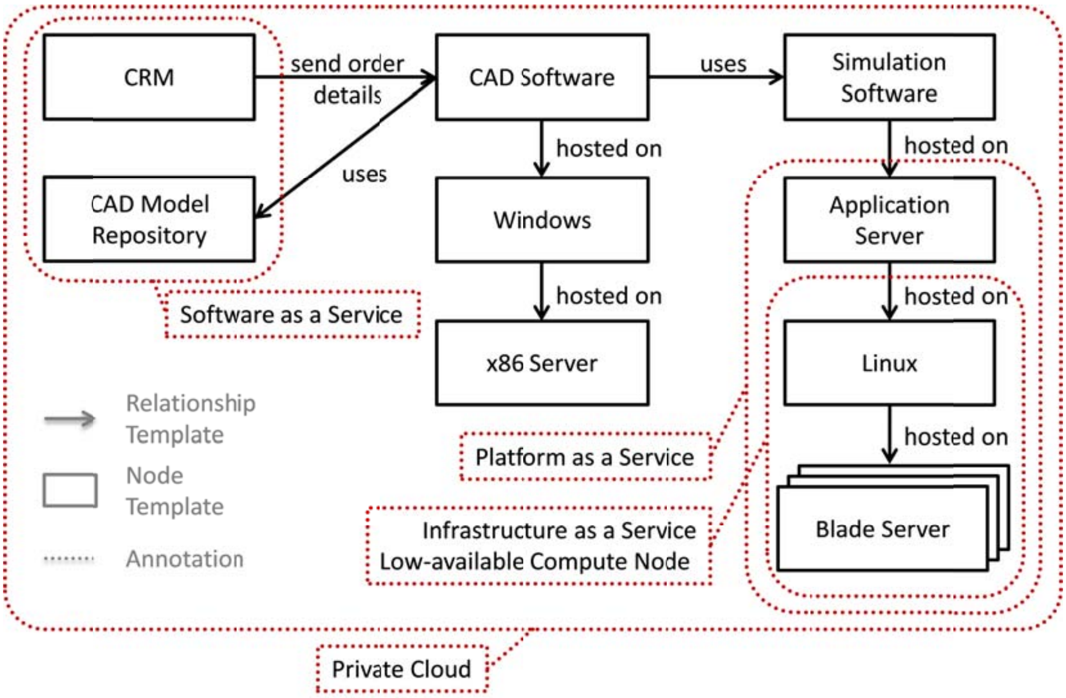

Fig. 10 TOSCA Service Template annotated with Cloud patterns

As described above, the business engineer recommended improving the *simulate model* activity which is realized by the *simulation software* application node in Fig. 10. When analyzing the application stack of the *simulation software* within the TOSCA Service Template the system architect recognizes that it mainly consists out of software components. Only the *blade server* on the bottom layer may be directly mapped to energy consumption and CO_2 emission. Of course, one need to keep in mind that software running on hardware components may also be responsible for inefficient resource consumption. However, these aspects are out of the scope of this work and we only focus on hardware components.

5.6 Migration

In Section 5.5 the System Architect identified the application components that need to be adapted. The decision to migrate these components is based on the chosen green business process patterns that are presented in the following.

Pattern 1: Resource Change Pattern

To realize the *Resource Change* pattern the architecture office decides to run their *simulation software* in a public cloud`. That means they move the application from their site to an external cloud provider. A cloud offering is chosen that guarantees at least the same computing power as the on-premise solution. Again, the objective addressed here is to reduce the CO_2 emission by using resources in a more efficient way or using more efficient resources in general. One example for an ecological-aware cloud offering is the Amazon Web Service Oregon Region where mostly hydro-electric power is used to operate the data centers [20]. This decreases the energy consumption and the corresponding CO_2 emission compared to conventional on-premise data centers of small and mid-sized enterprises significantly. For the implementation of the *Infrastructure as a Service* pattern with respect to the *Resource Change* pattern different alternative solutions are annotated. That means different TOSCA Service Templates are available. Fig. 11 shows one of the alternatives, featuring an Amazon EC2 instance in Oregon. A Node Type defines a set of properties from which a Node Template can choose a concrete implementation from.

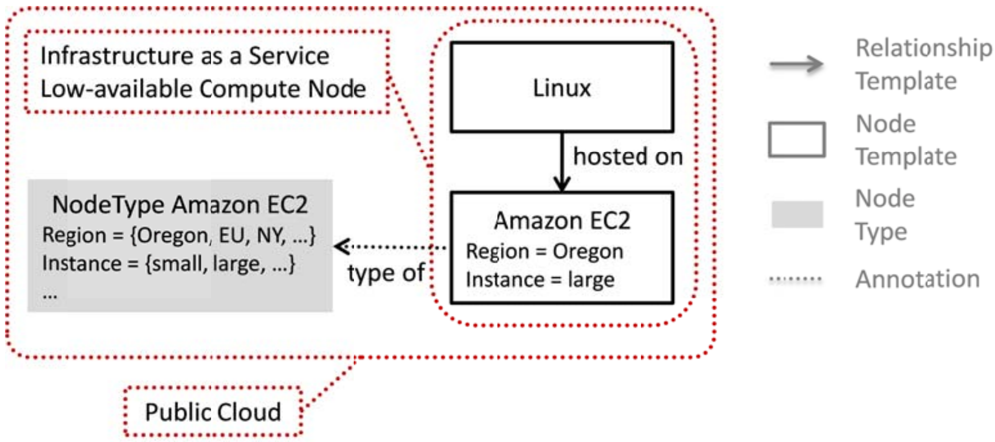

Fig. 11 Implementation of Resource Change pattern

Fig. 11 depicts that the 'Oregon' region and the instance size 'large' has been chosen for adapting the infrastructure of the simulation software (see Node Template 'Amazon EC2'). Another alternative is the definition of a Node Template that selects the 'New York' or 'Europe' region. However, those would emit more CO_2 and consequently have not been chosen. By exchanging the components that are annotated by *Infrastructure as a Service* and *Low-available Compute Node* in the TOSCA Service Template shown in Fig. 10 with the implementation shown in Fig. 11 the infrastructure layer of the application is changed to the new one.

The hosting of the components shown in Fig. 11 within a public cloud, however, is not that simple. When moving application components from a private Cloud to a public Cloud it may be necessary to secure the communication between these components and the associated components. The migration leads to a hybrid Cloud scenario: Parts of the infrastructure are hosted on-premise while the simulation environment is hosted in a public Cloud. To realize such a scenario, the pattern catalog with its pattern correlations guides system architects and developers to find and implement corresponding solutions.

In our example, they need to introduce a new component called *Cloud Computing Gateway*. Due to the restricted communication between the different environments, this component introduces additional application logic that ensures the communication between those environments.

Pattern 2: Green Compensation Pattern

Green Compensation is done for instance by donating money to a climate organization (e.g., NABU [21]) which, for example, plants new trees that are able to absorb CO_2. This will not directly reduce the CO_2 emission that is caused by the business process but compensates the emission from a global perspective. This pattern is realized by a TOSCA Service Template that consists of a Web service which the architecture office will host on Google AppEngine [22]. The TOSCA Service Template and the corresponding business process activity are shown in Fig. 12. The usage of this pattern changes the business process layer by providing an activity called *compensation service*. This activity needs to be provided for using the donation functionality within the business process. The application architecture layer is extended by integrating the TOSCA Service Template shown in Fig. 12.

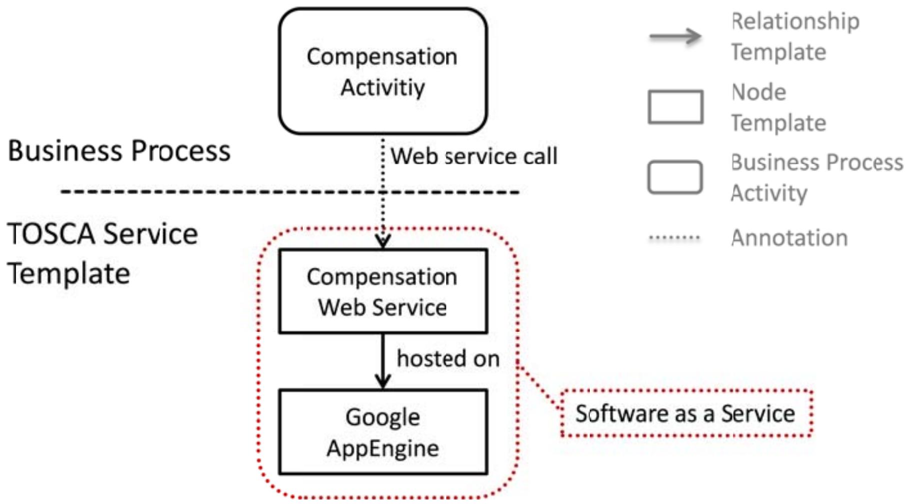

Fig. 12 Implementation of Green Compensation pattern

The architecture office has decided to apply both the *Resource Change* and the *Green Compensation* pattern. On the one hand, this reduces energy consumption of the compute nodes required for simulation and on the other hand, parts of the CO_2 emission of the complete process can be at least partly compensated. Fig. 13 depicts the adapted TOSCA Service Template as a result of our proposed approach. The infrastructure of the bridge construction process is now partly hosted in the public cloud.

Due to the changes in the hosting environment, the application is now annotated with the new pattern *hybrid Clouds*. This pattern allows the referencing of additional Cloud patterns that help the system architect and developers with finding and implementing solutions for specific problems occurring with this new application topology. One example is the relation between the Node Templates *CAD software* and *simulation software* that must be reestablished by using the *Cloud Computing Gateway pattern*.

This pattern is annotated to the relation between the *CAD software* and *simulation software* Node Templates. We further annotated the Node Templates, whose implementation is hosted in the public Cloud of Amazon and Google, with the *Public Cloud* pattern.

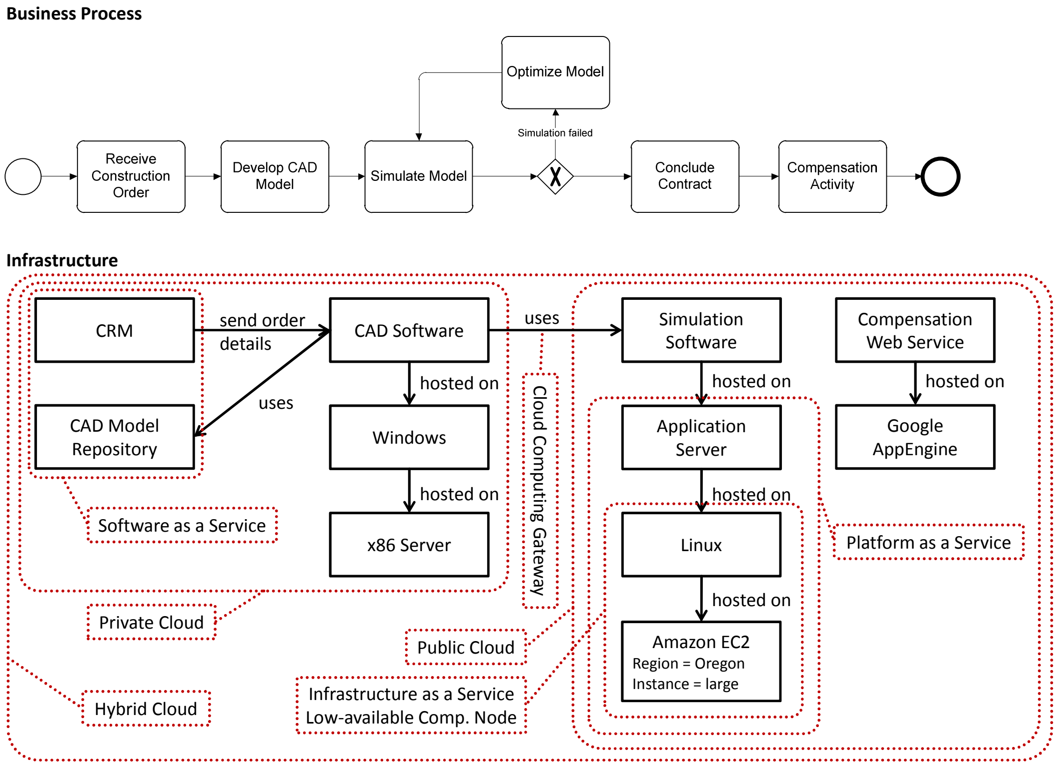

Fig. 13 Adapted TOSCA Service Template and the corresponding business process

5.7 Monitoring and Analysis

The previous sections have described the adaptation process up to the implementation of a new application topology that is underlying the business process. This adaptation has affected both the business process as well as the application topology. To verify the positive outcome of the adaptation the new environment needs to be further monitored and analyzed. On the one hand, this allows the evaluation of the adaptation actions and on the other hand, provides the basis for further adaptations. The most recent approach for supporting these tasks is described by Reichert et al. [30]. They use personalized visualizations to render monitoring information. As a result, the architectural office may decide at some point in time, for example, to move their *simulation environment* to a more sustainable cloud provider.

6 Related Work

The need to optimize business process and corresponding applications based on their environmental impact has been described in [4][5][6]. Many organizations already follow various approaches to decrease their environmental impact and to improve their image towards customers. These approaches observed in industry have been captured in the work of Nowak et al. [12] and are described as structured patterns. In this paper, we use these patterns as one possible example showing the feasibility of our abstract approach.

The model driven architecture (MDA) [31] proposes a generic approach for generating software out of models. When generating the platform specific model (PSM) out of the platform independent model (PIM), the platform model (PM) is annotated to the transformation and influences the nature of the transformation. In our work, we annotate patterns by TOSCA Service Templates and annotate Node Templates used in an existing application by Cloud patterns. The existing annotation is then manually transformed to a modified application using the Cloud patterns. There is no automatic transformation as MDA offers. Pignaton et al. [32] use MDA to generate QoS monitoring components out of a model annotated by QoS properties. They do not generate QoS properties into applications. In our work, we use annotations to select the best fitting TOSCA Service Template for a chosen pattern.

Wellhausen [33] explains how to discover, formalize, and write new patterns in general. In research there are approaches based on patterns which help to model new and reengineer existing business processes. Brahe et al. [34] use patterns to support the refinement of business processes. Analysts and architects describe processes in a domain specific language which is refined using patterns, supporting developers to create BPEL processes. Recurring transformations are captured in configurable patterns which define the information needed to execute the refinement of the process. Brinkmann et al. [35] use reusable patterns implemented as BPEL fragments to generate business processes controlled grid computing jobs. Our approach shows how sustainable reengineering patterns can be combined with cloud architecture patterns to improve the environmental aspects of business processes.

Weber et al. [36] present a set of generic process change patterns are described that focus on control flow adaptations (e.g. adding an activity to a process). Adaptations of processes based on QoS agreements and service level agreements (SLAs) are also discussed in Wetzstein et al. [37]. In contrast to the approach described here, the work in [37] focuses on aspect-oriented adaptation of processes during runtime to prevent possible SLA violations.

7 Conclusion and Outlook

This paper presented a method for adapting process-based applications. An overview of the method has been given in Section 1. The foundation of the method is a business process model that should be adapted based on certain criteria. In this work we refer to sustainable business processes. Thus, one or more *green business process patterns* are applied to the process model. A green business process pattern describes the adaptation actions in a high level manner. To support developers during the adaptation process adequately we have introduced a Cloud and green business process pattern correlation matrix and classification schema. The application of the *green business process patterns* is supported by a set of *Cloud patterns*. These Cloud patterns describe how to create and manage applications and services in a way supporting the ecological adaptation of the application. Both pattern models have been described in a uniform pattern format describing common solutions to reoccurring problems within the domains of eco-efficient business processes and cloud computing. Our method uses the Topology and Orchestration Specification for Cloud Applications (TOSCA) to describe the concrete implementation of applications. Such a template defines the application's infrastructure and how the infrastructure is setup and maintained. A set of available interoperable TOSCA Service Templates was then used to adapt the existing application towards a more sustainable one.

From a more general perspective, the method supports technical implementations and adaptations of application components used by a business process. However, it has to be noted, that we used *green business process patterns* in this work only as one possible example on how the proposed *Cloud patterns* can be utilized in order to achieve cross-cutting concerns combined with a "higher" objective. Other use-cases such as compliance or security may be applied as well.

The results of this work form the basis for a variety of aspects in future works. First, the general approach is intended to be used within a research project called Migrate! [9] that is funded by the German government (Grant ID 01ME11055). The approach allows us to provide the means for describing applications that can be migrated between different Cloud providers. The proposed pattern-based approach in combination with TOSCA eases the interoperability between these vendors. Another aspect we want to address in our future work is the derivation of best-practice solutions in form of new patterns. These patterns should provide solutions for specific problem statements by describing how to change business processes and their underlying software stack. Furthermore, we want to define a composition language for the use of patterns in architectural modeling languages such as the Architecture Description Language (ADL). This will be needed as the composition of annotated application models may be over-specified because annotated models may use the same middleware stack. Therefore, we need to avoid the redundancies and to support architects by providing an automatic consolidation of application models.

References

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

[2] Weske M (2007) Business Process Management: Concepts, Languages, Architectures. Springer-Verlag, Berlin Heidelberg

[3] Hammer M, Champy J (1994) Reengineering the Corporation: A Manifesto for Business Revolution. Reprint, HarperBusiness

[4] Ardagna D, Cappiello C, Lovera M, Pernici B, Tanelli, M (2008) Active Energy-Aware Management of Business-Process Based Applications. In: Proc. of the 1st European Conf. on Towards a Service-Based Internet. Springer-Verlag, Berlin Heidelberg, 183-195

[5] Hoesch-Klohe K, Ghose AK, Le LS (2010) Towards Green Business Process Management. In: Proc. of the IEEE Intl. Services Computing Conf. IEEE Computer Society, Washington, DC, USA, 386-393

[6] Nowak A, Leymann F, Schumm D, Wetzstein B (2011) An Architecture and Methodology for a Four-Phased Approach to Green Business Process Reengineering. In: Proc. of the 1st Intl. Conf. on ICT as Key Technology for the Fight against Global Warming. Springer-Verlag, Berlin Heidelberg, 150-164

[7] Schumm D, Leymann F, Streule A (2010) Process Viewing Patterns. In: Proc. of the 14th IEEE Enterprise Distributed Object Computing Conf. IEEE International, 89-98

[8] Reichert M, Kolb J, Bobrik R, Bauer T (2012) Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: 27th ACM Symposium on Applied Computing. ACM, accepted for publication

[9] Migrate! Research Project (2012) Modelle, Verfahren und Werkzeuge für die Migration in Cloud-basierte energieoptimierte Anwenderinfrastrukturen und deren Management. http://www.migrate-it2green.de/

[10] Fehling C, Leymann F, Mietzner R, Schupeck W (2011) A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures. Technical Report Nr. 2011/05, University of Stuttgart

[11] Fehling C, Leymann F, Retter R, Schumm D, Schupeck W (2011) An Architectural Pattern Language of Cloud-based Applications. In: Proc. of the 18th Conf. on Pattern Languages of Programs. ACM Digital Library

[12] Nowak A, Leymann F, Schleicher D, Schumm D, Wagner S (2011) Green Business Process Patterns. In: Proc. of the 18th Conf. on Pattern Languages of Programs. ACM Digital Library

[13] Topology and Orchestration Specification for Cloud Applications (TOSCA). http://www.tosca-open.org/

[14] Nowak A, Leymann F (2011) Nachhaltige Geschäftsprozesse. In: Gabler Wirtschaftslexikon Online, Wiesbaden: Gabler Verlag

[15] Object Management Group (2011) Business Process Model and Notation (BPMN). OMG Available Specification, Version 2.0

[16] OASIS (2011) Topology and Orchestration Specification for Cloud Applications (TOSCA). http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

[17] W3C (2004) XML Schema. http://www.w3.org/TR/xmlschema-0/

[18] Mietzner R, Unger T, Titze R, Leymann F (2009) Combining Different Multi-Tenancy Patterns in Service-Oriented Applications. In: Proc. of the 13th IEEE Enterprise Distributed Object Conf., IEEE International, 131-140

[19] Nowak A, Leymann F, Schumm D (2011) The Differences and Commonalities between Green and Conventional Business Process Management. In: Proc. of the Intl. Conf. on Cloud and Green Computing. IEEE, accepted for publication

[20] Datacenter Knowledge (2011) Amazon's Cloud Goes Modular in Oregon. http://www.datacenterknowledge.com/archives/2011/03/28/amazons-cloud-goes-modular-in-oregon/

[21] Nature and Biodiversity Conservation Union (2011) http://www.nabu.de/en/index.html

[22] Google App Engine (2011) http://code.google.com/appengine/

[23] Leymann F, Fehling C, Mietzner R, Nowak A, Dustdar S (2011) Moving Applications to the Cloud: An Approach based on Application Model Enrichment. In: Intl Journal of Cooperative Information Systems. World Scientific, 20(3)

[24] Binz T, Leymann F, Schumm D (2011) CMotion: A Framework for Migration of Applications into and between Clouds. In: Proc. of Intl. Conf. on Service-Oriented Computing and Applications. IEEE, 163-166

[25] Trummer I, Leymann F, Mietzner R, Binder W (2010) Cost-Optimal Outsourcing of Applications into the Clouds. In: Proc. of the 2nd Intl. Conf. on Cloud Computing. IEEE, 135-142

[26] Fehling C, Leymann F, Mietzner R (2010) A Framework for Optimized Distribution of Tenants in Cloud Applications. In: Proc. of the 3rd Intl. Conf. on Cloud Computing. IEEE, 252-259

[27] Mel P, Grance T (2009) The NIST Definition of Cloud Computing. National Institute of Standards and Technology

[28] Armbrust M et al. (2009) Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28, University of California, Berkeley

[29] Leymann F (2009) Cloud Computing: The Next Revolution in IT. In: Proc. of the 52th Photogrammetric Week. University of Stuttgart, Stuttgart

[30] Reichert M, Kolb J, Bobrik R, Bauer T (2012) Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: 27th ACM Symposium On Applied Computing (SAC'12). ACM Press

[31] OMG: MDA Guide Version 1.0.1. OMG Document Number: omg/2003-06-01

[32] Pignaton R, Asensio J, Villagra V, Berrocal J (2004) Developing QoS-aware component-based applications using MDA principles. In: Proc. of 2004 IEEE Intl. Conf. on Enterprise Distributed Object Computing. IEEE, 172-183

[33] Wellhausen T, Fiesser A (2011) How to write a pattern? A rough guide for first-time pattern authors. In: Proc. of EuroPLoP 2011. Hillside

[34] Brahe S, Bordbar B (2006) A Pattern-based Approach to Business Process Modeling and Implementation in Web Services. In: Proc. of the 4th Intl. Conf. on Service Computing. Springer Verlag, Heidelberg, 161-172

[35] Brinkmann A et al. (2008) Employing WS-BPEL Design Patterns for Grid Service Orchestration using a Standard WS-BPEL Engine and a Grid Middleware. In: *Parallel Computing* (01). ACC CYFRONET AGH, 4-11

[36] Weber B, Reichert M, Rinderle-Ma S (2008) Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. In: Data & Knowledge Engineering, 66(3). Elsevier Science, 438-466

[37] Wetzstein B et al. (2009) Monitoring and Analyzing Influential Factors of Business Process Performance. In: Proc. of the 2009 IEEE International Enterprise Distributed Object Computing Conference. IEEE, 141-150

All links were last followed on November 28, 2011.