Sustainability Information Model for Energy Efficiency Policies

Ana Carolina Riekstin, Bruno Bastos Rodrigues, Viviane Tavares Nascimento, Claudia Bianchi Progetti, Tereza Cristina Melo de Brito Carvalho, and Catalin Meirosu

The authors present the existing ways of representing policies, focusing on information models. They then propose SLIM, the Sustainability Information Model for Energy Efficiency Policies, an extension to the IETF Policy Core Information Model extension that allows unification of the management of green capabilities and protocols throughout the network.

Abstract

The need to manage the energy consumption of network infrastructure has been addressed by a significant body of work in recent years. In general, energy management capabilities were developed independently and optimized for particular network layers and node features. The interaction between multiple such green capabilities when deployed simultaneously, as well as potential interactions with other existing functionality such as quality of service functions, need to be managed transparently by the operators. We developed SLIM, the SustainabiLity Information Model for Energy Efficiency Policies, as an add-on to the IETF Policy Core Information Model Extension to allow unifying the management of green capabilities throughout the network. We illustrate the flexibility of our approach by presenting a use case and describing an energy management system where SLIM was used.

Introduction

Driven by the widespread adoption of smartphones and other devices such as sensors and gadgets connected to the Internet, the amount of electrical power consumed by networks is growing. In response, energy efficiency features are being deployed in the network to make the operation more sustainable. However, due to the diversity of vendors and legacy equipment, the use of energy efficiency features and their coordination in an automated fashion is a complex task for network operators.

Network operators aim to automate the deployment and operations of new services by increasingly relying on software abstractions and the use of programmatic control methods. Abstractions and virtualization simplify the design and provisioning of services by moving dedicated appliances to generic servers. The programmatic control supports management and automation, helping administrators in delivering new services faster. In this context, "a generic policy-based management model that can be used to express policies on top of arbitrary configuration data models is essential" [1].

Policy-based network management (PBNM) and energy efficiency capabilities have been studied in recent years for sustainability-oriented policies. Putting together the requirements for modeling policies and saving energy, Riekstin *et al.* [2] determined that policy refinement advances are needed for sustainability-oriented policies. Analyzing the existing solutions, the same authors proposed in [3] a method able to refine policies considering multiple abstraction levels and orchestrate different energy efficiency capabilities. One important component of the method is the information model, which is able to represent the different abstraction levels of sustainability-oriented policies. PBNM demands information models comprising business and system entities that can be implemented in an easy manner [4].

The Internet Engineering Task Force (IETF) defined an information model in RFC 3198 as an "abstraction and representation of the entities in a managed environment, their properties, attributes and operations, and the way that they relate to each other. It is independent of any specific repository, software usage, protocol, or platform" [5].[1] According to Agrawal [6], most policy systems do not have explicit information models defined, but they probably were somehow implicitly defined in the developers' minds.

The Policy Core Information Model (PCIM), later extended to the Policy Core Information Model Extensions (PCIMe) [7], defined policy rules in a vendor-independent way, supporting the definition of different levels of abstraction. It was based on the IETF/Distributed Management Task Force (DMTF) Core Information Model (CIM), a conceptual framework for the schema of the managed environment.

To represent sustainability-oriented policies, besides supporting traditional policies for authorization and obligation (e.g., access control and quality of service), an information model must be flexible enough to accept green metrics, time representation, or scenario conditions to enforce energy efficiency rules. Furthermore, it should comprise the modeling of actions that are specific to energy management, such as the *sleeping* and *performance adaptation* actions. It should also support modeling new types of metrics, such as the Watts per bits ratio along with the traditional performance metrics, and, in this way, support the study of the trade-offs between saving energy and maintaining performance.

[1] According to Agrawal *et al.* [6], "an information model is a structure for organizing information or data. For this reason, it is often called a data model." Many discussions exist on information vs. data model, as can be seen in RFC 3444. We stick with the traditional definition from the IETF.

Digital Object Identifier: 10.1109/MCOM.2016.1600222CM

Ana Carolina Riekstin was with the University of São Paulo and is now with the University of Quebec; Bruno Bastos Rodrigues, Viviane Tavares Nascimento, Claudia Bianchi Progetti, and Tereza Cristina Melo de Brito Carvalho are with the University of São Paulo; Catalin Meirosu is with Ericsson Research.

While PCIM and PCIMe support specification of policies, their existing classes do not have the specificity required for the green policies modeling. Although the existing models enable the definition and translation of business policies, the enforcement of their capabilities demands a different modeling approach. We are not aware of works in the literature that define a generic information model specifically aimed at energy management policies for telecommunications infrastructure.

In order to fill this gap, we define a Sustainability Information Model (SLIM)[2] specified using Unified Modeling Language (UML), allowing energy management policies to be expressed at different levels of abstraction. To model the policies at each level of abstraction, PCIMe [7] was used as the basis, and extended to comprise energy management aspects. In this work we revisit the main concepts and related works, considering energy efficiency capabilities, PBNM, and policy representations. We present the SLIM and its different levels of abstraction followed by a use case in order to discuss its benefits, limitations, and future work directions. Finally, we summarize the work, presenting our final remarks.

Background

To understand the main concepts and related works involved in the proposal of SLIM, we divide this section in two parts. The first part describes PBNM concepts, while the second part focuses on efforts toward policy representation using information models.

PBNM

PBNM supports dealing with the complexity and heterogeneity of systems, devices, applications, and networks. IETF RFC 3198 [5] defined policy according to two perspectives. The first defines policy as a goal, course, or method to guide decisions. The second view, taken from RFC 3060 [10], describes policy as a set of rules that enables the management, administration, and control of the network resources.

Strassner [9] established five levels of abstraction for policies: business, system, network, device, and instance levels. Altogether, they were referred to as the Policy Continuum. Carvalho *et al.* [10] applied the Policy Continuum to sustainability-oriented policies and proposed a methodological approach for refinement. The business level comprises service level agreements (SLAs), guides, and goals. The system level comprises sustainability and performance indicators. The network level includes metrics for network operations related to technologies. The device level comprises metrics for device operation. The instance level has variables applied to devices and their components.

PBNM and energy efficiency capabilities have been studied in recent years for sustainability-oriented policies, defined as policies "that manage energy efficiency features in the network" [3]. Seven requirements were identified for the refinement of sustainability-oriented policies [2]. The refinement method should ideally:

- Support the *translation* of high-level policies into network-level actions
- Take into account the *resources* in the network, including the energy efficiency capabilities available
- *Verify* that the refined policies meet the requirements of the original policies
- Detect and solve *conflicts* among policies
- Handle *dynamicity* by supporting different time slots or be able to determine what to do when the scenario changes
- *Orchestrate* different energy efficiency capabilities, that is, choose the best capability considering the network situation, combine capabilities in order to increase the energy efficiency, and avoid combining conflicting capabilities
- Be able to *represent policies* in order to keep the context, coherence, and integrity of the network

Recent advances in software defined networks (SDN) have led to extra features to address these requirements. It introduces more programmability and flexibility to the control plane through a centralized management point, aware of the whole network. This allows the development of more complex management techniques in an easier way compared to legacy networks.

Policy Representation

Several proposals have been developed to represent policies. The Policy Description Language (PDL) introduced semantics and an architecture using the event-condition-action form to define a policy as a function that maps a series of events into a set of actions [13]. Although not covering all classes of policies, the syntax of PDL was simple, and policies could be described as an implementation of the Extensible Markup Language (XML) syntax, which resulted in self-describing documents.

Ponder enforces obligation policies in a similar approach to PDL, and offered a complete toolkit to specify, analyze, and enforce policies. Its last version, Ponder2 [11], was an object-oriented runtime policy management framework for authorization and obligation policies. Policies in Ponder2 were defined using PonderTalk, a high-level language based on Smalltalk.

Using information models is another way to represent policies, as suggested by the IETF. It is typically represented using UML to represent managed objects at a conceptual level, modeling the structure, relationships, constraints, rules, and operations between managed objects and policies that affect their management [12]. Often, policies have the form *if Condition, then Action*, and could be represented in what is called a *condition-action policy information model*. One real-world extension of this model incorporates the event in the policy definition. The event can also be described in the condition part of the basic policy rule as *if Event and condition C is true, then Action* [6].

Extracting information from application-specific information models presents some benefits, such as easy modification when needed, policy reuse in different application domains by deploying a different information model, and standardized information representation. Further, Damianou [13] states that objects can be mapped to structure specifications, such as XML.

The Open Networking Foundation (ONF)[3] released its Core Information Model (ONF-

Using information models is another way to represent policies, as suggested by the IETF. It is typically represented using UML to represent managed objects at a conceptual level, modeling the structure, relationships, constraints, rules, and operations between managed objects and policies that affect their management.

[2] Essentially, SLIM is just one information model with different levels of abstraction, but we can call the different levels *information models*.

[3] The Open Networking Foundation (ONF) is an organization that promotes the development and adoption of software-defined networking.

To specialize PCIMe to sustainability-oriented policies, new classes were created to describe sustainability rules, conditions, and actions. The information model should also model the elements the administrator manipulates to compose the business rules.

CIM), which provides a representation of data plane resources for management control. According to the document, "the controller expresses a view of the network, in terms of ONF-CIM artifacts, to client SDN controllers/applications to meet the needs of that client." As a future work, they intend to develop a policy module.

PCIM and PCIMe: PCIM is an "object-oriented information model for representing policy information" [8]. It was developed as a complementary document of the CIM, an object-oriented information model published by the DMTF. In order to bring quality of service (QoS) requirements to the network, the specific QoS Policy Information Model (QPIM) was standardized by [14]. It specializes the PCIM to describe QoS actions. PCIM is not bound to a particular implementation; therefore, it can be used to exchange information in a variety of ways.

The model structure comprises two types of objects:
- *Structural classes*, which define ways of representing and controlling policy information
- *Associative classes*, which indicate how the class instances are related

In PCIM, a policy (class *Policy*) is defined by a set of rules (class *Policy Rules*), grouped by the *Policy Group* class. Each rule is composed of a set of conditions (class *Policy Condition*) and a set of actions (class *Policy Action*). The rules can also comprise time conditions (class *Policy Time Period Condition*). The actions can be executed in a specific order using the attribute *PolicyRule.SequencedAction*. Variables and values are used to build conditions following the structure (*<variable> MATCH <value>*) [7].

PCIMe was built on top of PCIM, and two main changes were introduced: the inclusion of new elements, extending PCIM to areas that it did not previously cover; and the update of deprecated elements, such as policy rule priorities, replaced by priorities tied to associations that refer to policy rules.

SUPA: Simplified Use of Policy Abstractions (SUPA) is under development at IETF [1]. Given that different industry actors embrace specific policy languages based on terminologies and concepts that are familiar to each specific technology domain, SUPA aims to define a model for expressing policies at different levels of abstraction, independent of languages, protocols, and data repositories. Rather than using multiple software for each policy language, it aims to provide a common information model addressing different representations.

SUPA is focused on addressing some PCIM shortcomings, such as class inheritance issues and the lack of design patterns that would help build hierarchies (SUPA uses the composite pattern), among others. We kept using PCIM and PCIMe for SLIM because SUPA was work in progress, and its new constructs would not address all gaps regarding the energy management policies that we need to describe. In the future, SLIM could be evaluated as a specialization of SUPA.

SLIM

As detailed in RFC 3640 [7], extensions of the defined information models may be defined, as QPIM extended PCIM for describing QoS policies. It states that properties can be included in the existing classes, while new classes and sub-classes can be defined [7] The policy description, enforced by the information model, may address anything starting from the business directives all the way to the detailed configuration of the nodes.

Overview

The PCIMe base classes that are present in the different SLIM abstraction levels are:
- PolicySet, used to group policies.
- PolicyRoleCollection, an addition in relation to PCIM, used to aggregate resources that share a common role.
- PolicyGroup, derived from PolicySet, which is a "container for a set of related PolicyRules and PolicyGroups."
- PolicyRule, inherited from PolicySet, which is the main class to represent the "If Condition then Action" semantics. It has two mandatory properties, SequencedActions and ExecutionStrategy, which are further defined by other classes aggregated by PolicyRule.
- PolicyCondition, an abstract class used to define policy conditions.
- PolicyAction, an abstract class used to define policy actions.
- SimplePolicyCondition, composed by the *<variable> MATCH <value>* structure.
- PolicyTimePeriodCondition, which represents the periods during which a PolicyRule is active.
- CompoundPolicyCondition, which aggregates simple policy conditions. This class has the property ConditionListType, used to specify if the associated policies are in disjunctive normal form (DNF), the default option, or conjunctive normal form (CNF).
- SimplePolicyAction, which models the structure *SET <variable>* to *<value>*.
- CompoundPolicyAction, which aggregates simple policy actions. This class has the properties SequencedActions and ExecutionStrategy, reported to the PolicyRule class.
- PolicyValue, an abstract class to define value objects. Examples of derived classes are PolicyIPv4AddrValue and PolicyMACAddrValue.
- PolicyVariable, an abstract class used to define variables, which are used in individual conditions.
- PolicyExplicitVariable, which "indicates the exact model property to be evaluated or manipulated."
- PolicyImplicitVariable, which models "implicitly defined policy variables (that) are evaluated outside of the context of the CIM Schema." Sub-classes must specify the data type and semantics of the variables [7].

We refer the reader to RFC 3460 [7] for more details on the classes and sub-classes.

To specialize PCIMe to sustainability-oriented policies, new classes were created to describe sustainability rules, conditions, and actions. The information model should also model the elements the administrator manipulates to compose the business rules (user, SLA, class of service,

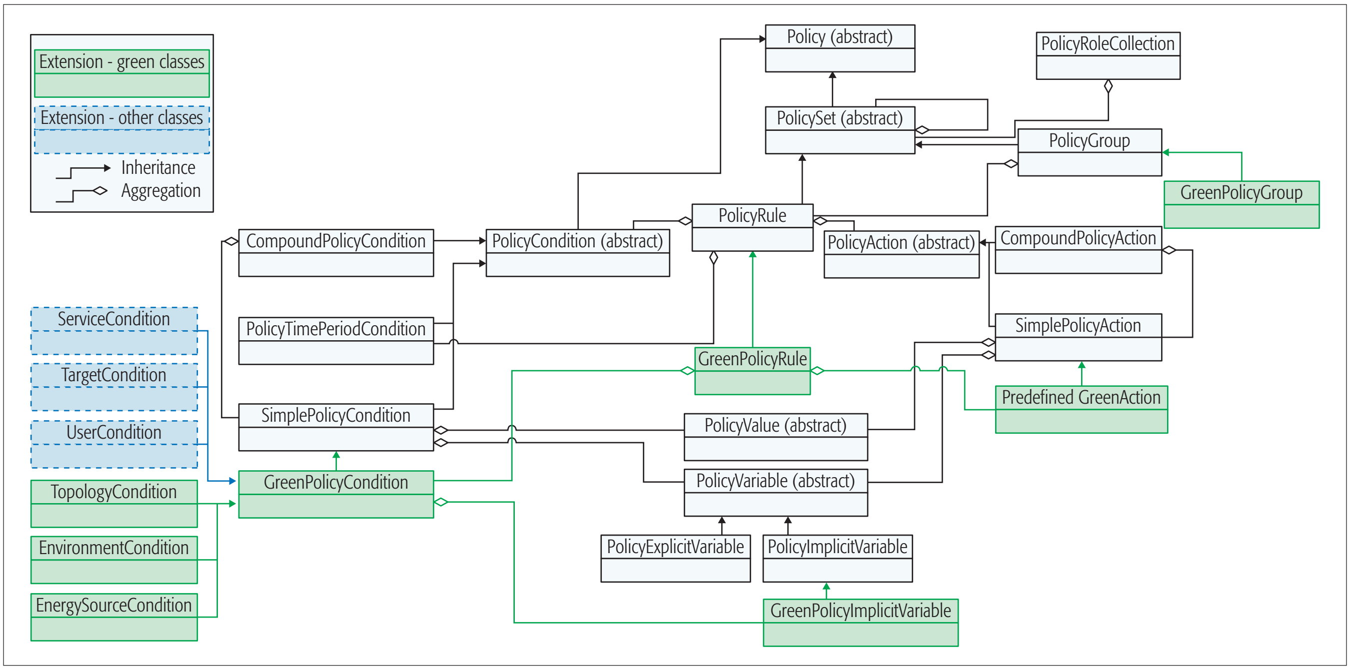

Figure 1. SLIM — business/system level of abstraction.

application, servers), and the information needed to configure the devices [15].

Considering the sustainability-oriented policies' requirements, the different Policy Continuum levels, the PCIM/PCIMe characteristics, how QoS can be handled by QPIM and the extension proposed by Beller *et al.*, and how the SLA semantics was supported by them [15], three abstraction levels are proposed for SLIM: business/system, network, and device levels for sustainability-oriented policies representation. The Instance Level Information Model is particular for each technology or vendor, and therefore was not detailed.

The business and system information models are the same — they completely represent both levels with the same classes. They are still considered as two different layers to ensure the generality necessary for the near future systems. This first SLIM level is further specialized in each of the other levels. The Network Level Information Model is directly influenced by QPIM and includes technology-specific, device-independent information. The Device Level Information Model is the most detailed and considers device-specific information, including the variables that are expected to be managed to put the green capabilities to work.

Business and System Levels

Figure 1 details the business and system information model level. The *GreenPolicyRule* derives from the PolicyRule and is used to determine the conditions and actions for sustainability-oriented policies.

The SimplePolicyCondition is extended by the *GreenPolicyCondition*. The latter is extended by the *EnvironmentCondition*, which relates to the current situation in the network, such as the traffic being handled by the network, and by the *EnergySourceCondition*, which relates to the influence of different energy sources on the policies (e.g., for green SLAs that limit the amount of carbon to be emitted). The GreenPolicyCondition is also extended by the *TopologyCondition*, which represents the different topologies for the network, since specific topologies (e.g., SoHo, WAN, fat tree) can trigger different actions (e.g., the elastic tree capability works only for fat trees).

The other classes inheriting from the GreenPolicyCondition are supporting classes to represent users, services, and any associated condition (e.g., users in the HR department).

The SimplePolicyAction is extended by the *Predefined GreenAction* in the Business/System Levels Information Model, which determines different green plans for users under different contracts (Green SLAs). This is the part that changes more in the Network Level Information Model and the Device Level Information Model in relation to the Business and System Levels.

The Variables are extended by the *GreenPolicyImplicitVariable* class, which is further extended to comprise:
- At the business level, for instance, the operational expenditures (OPEX) accounting or total carbon emissions for Green SLAs
- At the system level, Watts per hour per energy source or the power usage effectiveness (PUE) facility, among others

Network Level

Figure 2 details the network information model level. This level is related to the capabilities that act on the whole network, such as those related to green traffic engineering. This level also needs to be aware of the capabilities that act in the device and its components to ultimately enforce the rules.

The network level information model details the business and system classes, giving policies a more technical format, with configuration details, represented in the class *Config GreenPolicyRule*.

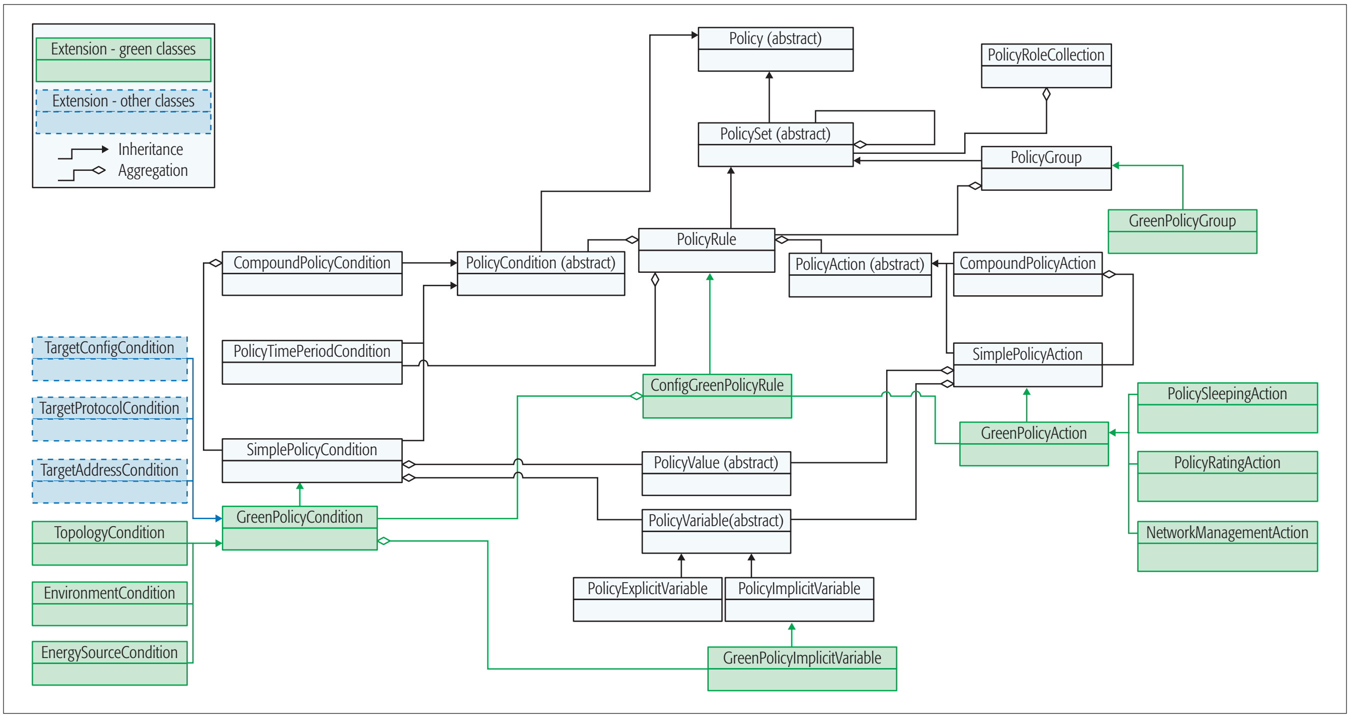

Figure 2. SLIM — network level of abstraction.

This abstraction level groups the conditions for the capabilities employment on the *GreenPolicyCondition* class. The defined conditions suggest the configuration of protocols and addresses components susceptible to the application of energy efficiency features. These classes are *TargetConfigCondition*, *TargetProtocolCondition* and *TargetAdressCondition*.

These supporting classes demand the definition of different PolicyValues, comprising, for instance, a value for the traffic load. The *GreenPolicyImplicitVariable* also gets a more technical aspect, representing, for instance, maximum loads for the whole network, or the Watts/bits ratio.

In the Action part, the business and system goals take the format of a more technical *GreenPolicyAction*. The Sleeping and Rating types of actions appear, detailing the approach each policy requires to put the higher-level policies into practice. *PolicyRatingAction* checks which devices are below or above the load levels defined by the user for each link rating (e.g., 10 Mb/s, 100 Mb/s, and 1 Gb/s), while the *PolicySleepAction* changes the device state when it reaches the specified load level of the system policies. The *NetworkManagementAction* class models the capabilities that are aware of the entire network to make decisions such as those related to green traffic engineering (followed by sleeping actions to save energy on underutilized nodes), or to virtual machines migration.

Device Level

Figure 3 details the device information model level. This level is related to the capabilities that act in the node, but also deals with the device components. The instance level policies will ultimately be applied on the node components and capabilities. The device level details the conditions and the device-specific actions, naming the variables that each green capability must handle. It models, in a generic way, the possible energy efficiency actions in a device: sleeping and rating. The SimplePolicyAction is extended by the *GreenPolicyAction*, which determines different policies related to energy efficiency capabilities: *PolicySleepingAction* and *PolicyRatingAction*. These classes can then be extended to comprise information for the capabilities available in the network, using *DeviceSleepingAction* and *DeviceRatingAction*.

There are also supporting classes that describe the conditions of the device and its components, which are *TargetDeviceCondition*, *TargetConfigCondition*, *TargetProtocolCondition*, and *TargetAddressCondition*. These supporting classes demand the definition of different PolicyValues, comprising, for instance, the TargetDevice ID, its interfaces, power profile, and maximum and current load it can handle. The *GreenPolicyImplicitVariable* also goes down to the device level, representing, for instance, the node Energy Consumption Rating (ECR) or its Energy Proportionality Index (EPI), which can be used inside the conditions in this level. Table 1 summarizes the extensions.

This section has presented the different levels of abstraction proposed. As shown, all the enforcement aspects of the energy management capabilities were included in the SLIM, ranging from the conditions to the actions considered. SLIM enables an independent specification of the refinement functionality associated to policy translation between the levels.

Use Case

To illustrate how SLIM works, we briefly describe the orchestration method proposed by Riekstin *et al.* [3], in which SLIM was used. Figure 4 depicts the method implementation architecture. The method uses table lookups for the high-level policies translation, with the support

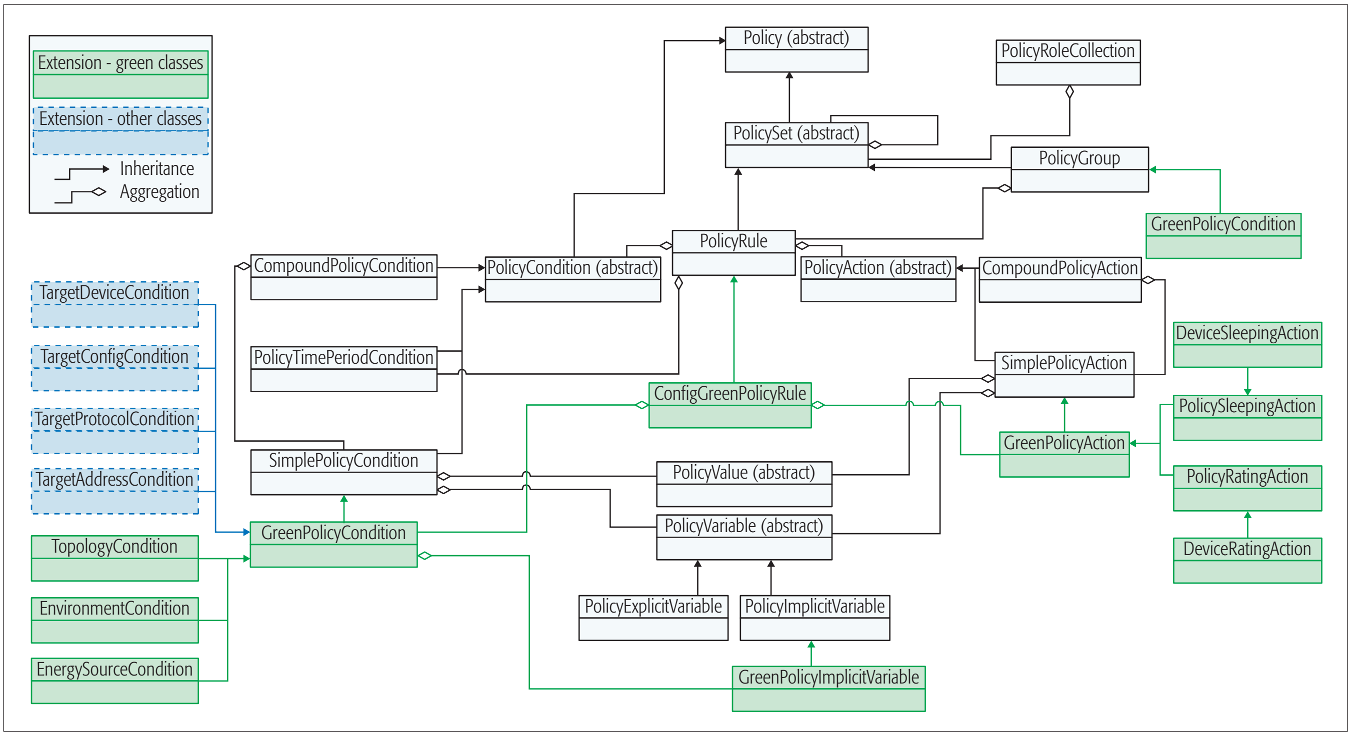

Figure 3. SLIM — device level of abstraction.

of SLIM to represent the policies. Translated policies are then used in conjunction with a utility function to deploy a decision tree able to select the best energy efficiency capability (or combination of capabilities) for a given scenario, ensuring conflict-free operation and a capability(-ties) selection that will not lead to congestion or high packet losses.

A set of tables needed to be predefined, related to Environment Condition, Time Condition, and Action. The *Environment Condition* was used to model the high or low use of the capacity of the network structure: the input will determine the traffic load conditions for the use of the energy capabilities. Table 2 illustrates the data that can be defined as environment conditions. Each row represents one possible environment condition, while the column represents the translated policy from one level to the other, using the information from SLIM.

The *Time Condition* is the input that will provide the data regarding the period of the day when the energy capabilities can be applied to the network infrastructure. Table 2 presents the time conditions. Each row represents one possible time condition, while the column represents the translated policy from one level to the other, using SLIM.

The last input is the *Action*. It determines the energy efficiency behavior that must be applied in the network infrastructure or if the performance of the network is more important than the reduction of the electrical expenditure at a given point of time. Table 2 shows the data that can be defined as actions. Each row represents one possible action, while the column represents the translated policy from one level to the other, using the information from SLIM.

Each one of the business policies received is compared to the content of the respective table.

Extension	Summary
Business/System level	
Green policy rule	Has the conditions and actions for sustainability-oriented policies
Green policy condition	Sustainability-oriented conditions: Topology, Environment, and EnergySource conditions representation
Green policy implicit variable	Represents the energy-related variables, such as Watts/bits ratio
Predefined green action	Represents different green plans for the users
Network level	
Config green policy rule	Conditions and actions in a more formal manner, such as event-condition-action policies
Green policy condition	Conditions get more technical format, such as representing the network total capacity and protocols
Green policy implicit variable	Supporting the conditions, also gets a more technical aspect, for instance, comprising the maximum load the network can handle
Green policy action	Represents management actions (whole network) and is aware of node actions (sleeping and rating)
Device level	
Green policy condition	Conditions go to the devices, detailing, for instance, IDs, their roles (core/edge), etc.
Green policy implicit variable	Variables related to the devices, such as device indexes
Green policy action	Represents device actions, grouped in rating and sleeping (can be extended to comprise others in the future)

Table 1. Examples of policy refinement using SLIM: business to system to network to device level.

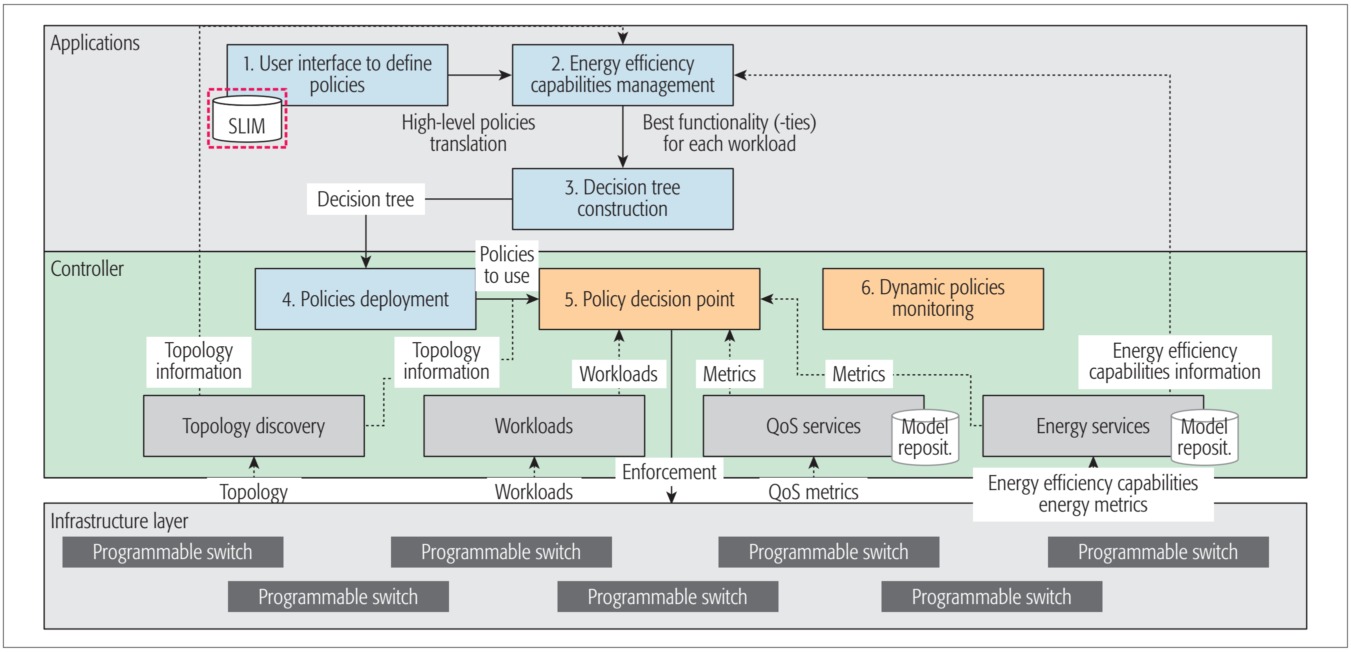

Figure 4. Orchestration method implementation modules with the SLIM highlighted [3].

The translated information is selected from the table, and is saved in a repository for future use.

To help understand how table lookup is employed by the method, an example of information defined in the Interface and its respective translation is provided below. An example of business policies defined in the interface is:

Environment Condition: *if usage is low*
Time Condition: *during the night*
Action: *save energy*

The module opens the *Environment Condition — Business to System Level* table, searches for the *if usage is low* input, and then selects the percentage of the load for the condition. The module then opens the *Environment Condition — System to Network Level* table and performs the same operation to translate the percentage to Mb/s information.

if usage is low → 20% → NetworkCapacity = 1 Gb/s and Load < 200 Mb/s

The module opens the *Time Period Condition* table, searches for the *during the night* input, and then selects the time (start and end time) that the method can be employed.

during the night → 22 h < hour < 6 h → starting time: 22 h, ending time: 6 h

The module opens the *Action — Business to System* table, searches for the *save energy* input, and then selects the action required for the action. The module performs the same to translate from the system to the network level of the Policy Continuum,

save energy → use energy efficiency in the network → check in the decision tree the best capability for the given bandwidth utilization, all capabilities available are allowed

The result is the metrics definition in a repository, already translated, and ready for the use by the orchestration part of the method:

['NetworkCapacity==1000 and load < 200.0', 'time > = 22 and time < = 6', check in the decision tree the best capability for the given bandwidth utilization, all capabilities available are allowed']

For instance, the available capabilities are related to green traffic engineering followed by putting the unused nodes to sleep. Therefore, on the network level, the policy will allow the green traffic engineering capability and enforce it, using the representation of the Network Management Action in SLIM. On the device level, the policy will tell the system to put unused nodes to sleep, using the representation of the Device Sleeping Action in SLIM.

Discussions and Future Work Directions

Defining information models focused on sustainability takes a step toward a common representation of green policies. This work extends PCIMe to support specific requirements and constraints that sustainability-oriented policies require. SLIM was defined with different levels of abstraction to be aligned to the Policy Continuum and thus operate at several levels of abstraction. SLIM was implemented and validated as part of a method that was previously published [3].

These new classes introduced by SLIM, however, do not restrict the policy definition for new capabilities that may be proposed: this is because SLIM was defined generically, not restricting the green classes to existing technologies. If required, new sustainability aspects could be included, such as life cycle assessment. Regarding the actions, currently, they can be related to sleeping or rating, so they can be modeled using SLIM proposed classes. If new types of energy management capabilities are

Business level	System level	Network level	Device level (1..N)
Environment condition			
"If usage is low"	20%	NetCapacity = 1 Gb/s and Load < 200 Mb/s	DevCapacity = 100 Mb/s and DevLoad 20 Mb/s
"If usage is high"	80%	NetCapacity = 1 Gb/s and Load < 800 Mb/s	DevCapacity = 100 Mb/s and DevLoad 80 Mb/s
"In any condition"	Always	NetCapacity = 1Gb/s and Load = ANY	DevCapacity = 100 Mb/s and DevLoad = ANY
Time condition			
During the night	22h hour 6h	Start: 22h/End: 6h	Start: 22h/End: 6h
During the morning	6h hour 12h	Start: 6h/End: 12h	Start: 6h/End: 12h
During the afternoon	12h hour 18h	Start: 12h/End: 18h	Start: 12h/End: 18h
During the evening	18h hour 22h	Start: 18h/End: 22h	Start: 18h/End: 22h
During the day	6h hour 22h	Start: 6h/End: 22h	Start: 6h/End: 22h
Action			
"Save energy"	"Use energy efficiency in the network"	"Check in the decision tree the best capability for the given bandwidth utilization, all capabilities available are allowed. E.g., use green traffic engineering to concentrate traffic	"Enforce the selected green capability in each device. E.g., apply Synchronized Coalescing (sleeping capability) in all nodes or put unused nodes to sleep (after green traffic engineering)"
"Prioritize performance"	"Provide maximum QoS"	"Don't apply any energy effic. capability"	"Keep devices working at full performance"
"Save energy without reducing perform."	"Use energy effic. without reducing QoS"	"Apply only link rating capabilities"	"Enforce link rating capability(-ties) in each device"

Table 2. Examples of policy refinement using SLIM: business to system to network to device level.

developed, new classes could be added to support them, extending SLIM. The model presented in this work supports the refinement through the levels as shown with the use case presented.

SLIM addresses specifically the requirement *"Be able to represent policies in order to keep context, coherence, and integrity of the network,"* one of the requirements a method should fulfill to be able to refine and orchestrate energy efficiency capabilities [2]. Despite being specifically tied to this requirement, it is totally aligned with the others, supporting the translation, resources representation, verification, conflicts detection and resolution, dynamicity, and orchestration. For instance, the time dynamicity is only possible with the information model class *TimePeriodCondition*. Another example is the resources representation support, with the different levels of abstraction representing capabilities, nodes, and the whole network.

The information modeling exercise done for PCIMe can also be done for the cloud computing context with the OpenStack Congress, for instance, so that the framework is able to model energy efficiency capabilities using specific constructs. The SLIM for cloud computing environments must take into account, besides business level policies, energy efficiency capabilities, the different resources in the infrastructure (compute, storage, network), and the management of virtual and physical resources. Another interesting future work can emerge from the ONF-CIM initiative of defining a Policy Module in the near future.

Final Considerations

Energy efficiency is presented as an environmental and economical advantage for network operators since it can reduce the operational costs and the emission of pollutant gases. Many capabilities and protocols have been proposed to improve energy efficiency in networks. PBNM associated with information models help to manage the network infrastructure considering the availability of such capabilities and the traditional QoS and access control constraints, besides enabling a more automated operation.

Although different approaches for improving policy information models have been proposed, green policies were not considered previously. In this work we present SLIM, a sustainability-oriented information model with different levels of abstraction to fill this gap. We extend PCIMe including new groups and rules of green policies, as well as new conditions and green actions. Through the information model proposed, we hope to bridge the gap in order to improve the control of network energy efficiency, addressing one important part of sustainability-oriented policies refinement.

References

[1] J. Strassner, J. M. Halpern, and J. Coleman, "Generic Policy Information Model for Simplified Use of Policy Abstractions (SUPA)," IETF, Internet-Draft draft-strassner-supa-generic-policy-info-model-03, Jan. 2016, work in progress, accessed 25 July 2016; https://tools.ietf.org/html/draft-strassner-supa-generic-policy-info-model-03.

[2] A. Riekstin *et al.*, "A Survey of Policy Refinement Methods as A Support for Sustainable Networks," *IEEE COMST*, vol. 18, no. 1, 1st qtr. 2016, pp. 222–35.

[3] A. C. Riekstin *et al.*, "Orchestration of Energy Efficiency Capabilities in networks," *J. Network and Computer Applications*, vol. 59, Jan. 2016, pp. 74–87.

[4] S. Abeck, *Network Management: Know It All*, Morgan Kaufmann, 2009.

[5] A. Westerinen *et al.*, "Terminology for Policy-Based Management," IETF RFC tech. rep., Nov. 2001, accessed 25 July 2016; https://www.ietf.org/rfc/rfc3198.txt.

[6] D. Agrawal *et al.*, *Policy Technologies for Self-Managing Systems*, Pearson Education, 2008.

[7] B. Moore, "RFC3640 – Policy Core Information Model (PCIM) Extensions," IETF RFC 3460, Jan. 2003, accessed 25 July 2016; http://www.ietf.org/rfc/rfc3460.txt.

[8] B. Moore *et al.*, "Policy Core Information Model (PCIM) Extensios," ietf RFC 3060, Feb. 2001, accessed 25 July 2016; http://www.ietf.org/rfc/rfc3060.txt.

[9] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation*, 1st ed., Morgan Kaufmann, Sept. 2003.

[10] T. C. M. B. Carvalho *et al.*, "Towards Sustainable Networks – Energy Efficiency Policy from Business to Device Instance Levels," *ICEIS '12*, SciTePress, 2012, pp. 238–43.

[11] K. P. Twidle *et al.*, "Ponder2: A Policy System for Autonomous Pervasive Environments," *5th Int'l. Conf. Autonomic and Autonomous Systems*, Valencia, Spain, 20–25 Apr. 2009, pp. 330–35.

[12] S. Davy, B. Jennings, and J. Strassner, "The Policy Continuum-Policy Authoring and Conflict Analysis," *Comp. Commun.*, vol. 31, no. 13, Aug. 2008, pp. 2981–95.

[13] N. Damianou, *A Policy Framework for Management of Distributed Systems*, Ph.D. dissertation, Imperial College London, 2002, accessed 25 July 2016; http://pubs.doc.ic.ac.uk/policy-framework-distrib-systems/

[14] Y. Snir *et al.*, "Policy Quality of Service (QoS) Information Model," Nov. 2003, accessed 25 July 2016; http://tools.ietf.org/html/rfc3644

[15] A. Beller, E. Jamhour, and M. Pellenz, "Defining Reusable Business-Level QoS Policies for DiffServ," *Utility Computing*, ser. Lecture Notes in Computer Science, A. Sahai and F. Wu, Eds. Springer Berlin Heidelberg, vol. 3278, 2004, pp. 40–51.

Biographies

Ana Carolina Riekstin [M] (a.ca.riekstin@ieee.org) is a postdoctoral fellow at Synchromedia Laboratory, École de Technologie Supérieure, Montreal, Canada. She received her Ph.D. (2015) and M.Sc. (2012) in computer engineering from the Polytechnic School of the University of São Paulo (USP). She got her B.Sc. in computer science (2007) from the Institute of Mathematical and Computer Sciences, USP. She worked previously at the Laboratory of Sustainability in ICT (LASSU) at USP, Univesp, Microsoft Research (internship), Volkswagen do Brasil, and PromonLogicalis.

Bruno Bastos Rodrigues (brodrigues@larc.usp.br) has a B.Sc. in computer science from the University of the State of Santa Catarina (UDESC) in 2013. At the moment he is an M.Sc. candidate and researcher at LASSU in the Polytechnic School, USP. During his M.Sc., he has co-authored three patent applications in the area of network management, service virtualization, and energy efficiency.

Viviane Tavares Nascimento (vianetn@larc.usp.br) is an M.Sc. candidate and researcher at LASSU, Polytechnic School, USP, currently working on the Energy Efficiency Cloud (E2C) project. She graduated as an electrical engineer in 2008 from Universidade Estadual Paulista, Engineeromg Faculty of Ilha Solteira, and received her M.B.A. degree at the Laboratory of Network Architecture (LARC) of the Polytechnic School, USP. She worked from July 2008 to January 2013 at Itau-Unibanco Bank.

Claudia Bianchi Progetti (cprogetti@larc.usp.br) is a Ph.D. candidate at LASSU, Polytechnic School, USP. She received her M.Sc. in management and technology in production systems in 2014, her M.B.A. in business management with emphasis in project management in 2008, and her post graduate degree in systems analysis for applications and web solutions in 2005. Her career has developed in the area of information technology, with experience in large corporations in strategic and leadership positions.

Tereza Cristina Melo de Brito Carvalho (terezacarvalho@usp.br) is an ssociate professor and director of LASSU at the Polytechnic School, USP. She received her B.Sc. (1980), M.Sc. (1988), Ph.D. (1996), and free-docency (2012) in electronic engineering from USP. She concluded her Sloan Fellows Program (2002) as postdoctoral work at MIT. Her specialization is in local network projects in the Alfried Krupp von Bohlen und Halbach Foundation, Nürnbergm Germany (1990). She has more than 90 scientific and technology publications.

Catalin Meirosu (catalin.meirosu@ericsson.com). Is a master researcher with Ericsson Research, Stockholm, Sweden, which he joined in 2007. He holds a Ph.D. in telecommunications (2005) from Politehnica University, Bucharest, Romania, and was a project associate of the ATLAS experiment at the Large Hadron Collider at CERN, Geneva, Switzerland during his Ph.D. He has 10 granted patents and has co-authored over 30 scientific papers. He has been involved in several research projects on SDN, cloud computing, and energy management.