A Survey of Policy Refinement Methods as a Support for Sustainable Networks

Ana Carolina Riekstin, Guilherme Carvalho Januário, Bruno Bastos Rodrigues, Viviane Tavares Nascimento, Tereza Cristina Melo de Brito Carvalho, and Catalin Meirosu

***Abstract*—Green sustainability-oriented features have become common in network nodes and protocols. Running a network in an energy-efficient way is an important concern of network operators and datacenter networks. The implementation and coordination of the myriad of existing network features poses a challenging task. The energy efficiency capabilities must be selected according to the network conditions and can be combined to increase energy savings. However, they can conflict if not orchestrated in a proper manner. Policy-Based Network Management is a well-known approach to addressing the complexity of network management tasks. In conjunction with a refinement process to translate high-level policies down to low-level policies, it can bring business directives to the network, including sustainability goals. In this survey, we identify the major characteristics of sustainability-oriented policies, as well as the requirements a policy refinement method for such type of policies has to fulfill, including energy efficiency capabilities orchestration. We then analyze existing policy refinement techniques and discuss the challenges on how they address or need to be modified in order to be applicable to sustainability-oriented policies.**

***Index Terms*—Networks, policy refinement, management, sustainability, energy efficiency.**

I. Introduction

RUNNING a network in an energy-efficient way is an important concern of network operators [1]. This is triggered, on the one hand, by the increased awareness of issues related to sustainability among customers, and, on the other hand, by potential savings in terms of operating expenditures. Many solutions that address energy efficiency as the main operational aspect of sustainability have been developed to exploit the idleness of links and nodes. Such solutions are capabilities also referred to as sustainability, energy efficiency, energy savings, or green capabilities. According to Bilal *et al.* [2], energy efficiency can be achieved by concentrating traffic to put unused devices to sleep or by scaling down the link data rate.

Such solutions are as local as chip-level enablers for a more power-efficient node or as broad as routing protocols. Two typical local solutions are the use of the different power states as specified by the Advanced Configuration and Power Interface (ACPI) [3] and the use of Adaptive Link Rate (ALR) [4]. Energy-Aware Routing as Green Open Shortest Path First (OSPF) [5], [6] is an example of a protocol that aims to reduce energy consumption by synchronizing the behavior of nodes across the network. SustNMS [7] is an example of a solution combining green traffic engineering plus sleeping the unused devices. There are also standardization efforts, such as the Green Abstraction Layer (GAL), which proposes an abstraction layer to exchange information about power management settings of energy efficiency capabilities [8].

Seen in a generic manner, node-level energy-saving capabilities operate by evaluating the need for resources during a particular time interval and then automatically adjusting the system performance (thus potentially reducing the power consumption) to match the demand. Network-level energy-saving capabilities try to dynamically concentrate or re-route traffic such that particular paths could be transitioned towards a sleep mode either through commands from the management system or by automatic node-level energy efficiency capabilities. These capabilities were developed to operate independently from each other. Putting more than one to work on the same network could bring higher energy savings than enabling them separately, but could lead to conflicts. Such conflicts reduce or negate energy savings, or lead to undesired behavior, such as repeatedly turning on or putting a node to sleep.

Turning the networks greener also presents some other important challenges. Bilal *et al.* [9] discussed the issues on datacenter networks energy efficiency: in average, such infrastructures do not operate over 25% of the peak load, and the links remain idle about 70% of the time. As there are many scattered short idle periods rather than few and long periods, it is important to consider the traffic characteristic prior to applying energy saving capabilities. Besides, performance and reliability should be taken into consideration since degradation, latency, and reliability decrease could reduce revenues substantially.

As the complexity in such scenario increased, the use of policies to enforce network quality of service (QoS) or control access became more necessary [10]. Policies can be used to regulate the operation of such capabilities in order to avoid conflicts among them. Policy-Based Network Management (PBNM) is a well-known approach that can comprise either QoS, access control or sustainability-oriented policies, helping to address such management issues. PBNM presents several benefits, such as less manual countermeasures and errors, automated analysis and verification based on a formal foundation,

Manuscript received December 15, 2014; revised June 5, 2015; accepted July 24, 2015. Date of publication August 3, 2015; date of current version January 27, 2016. This work was supported by Ericsson Telecomunicações S.A., Brazil.

A. C. Riekstin, G. C. Januário, B. B. Rodrigues, V. T. Nascimento, and T. C. M. de Brito Carvalho are with University of São Paulo, 05508-070 São Paulo, Brazil (e-mail: carolina.riekstin@usp.br; gcjanuario@usp.br; tereza.carvalho@usp.br; brodrigues@larc.usp.br; vianetn@larc.usp.br).

C. Meirosu is with Ericsson Research, 164 83 Stockholm, Sweden (e-mail: catalin.meirosu@ericsson.com).

Digital Object Identifier 10.1109/COMST.2015.2463811

dynamic inspection and adaption at runtime, without demanding changes in the underlying implementation [11].

However, using policy-based management is not straightforward. The operators define the high-level policies, and these policies have to be refined into policies at the operational level, a process known as *Policy Refinement* [12]. A fully automated refinement of high-level policies down to low-level policies, applicable in the network, is still an open issue. The challenge increases if we consider the peculiarities of sustainability-oriented policies and that the policy refinement and enforcement is not only related to translation, but also to another requirements, such as policy analysis (coverage and conflicts detection and resolution), or resources discovery.

The existing policy refinement methods focus on translating policies from a high to a lower level of abstraction. In general, they support policy analysis, resources discovery and network dynamicity aspects. However, the methods neither fully support energy efficiency capabilities representation, nor allow the capabilities orchestration, that is, to coordinate and combine them to save more energy while ensuring a conflict-free operation.

This survey presents the necessary background to substantiate the challenges of sustainability-oriented policy refinement. Three main contributions are presented in this work: the definition of what is a sustainability-oriented policy and its different abstraction layers; a list of requirements to evaluate a policy refinement method considering sustainability issues; and a classification and analysis of policy refinement methods to support sustainability-oriented policies, showing the trade-offs between automation and the solution generality (regarding if it is domain specific or not).

To the best of the authors' knowledge, this is the most complete evaluation of refinement methods, as well as the first to evaluate them considering sustainability-oriented policies. Boutaba *et al.* [10] presented a historical perspective on policy-based management, but not focusing on policy refinement. Phan *et al.* [13] evaluated five policy frameworks (IETF, Ponder, KAoS, Rei and WS-Policy) in light of Service-Oriented Architecture (SOA)-specific criteria, but they did not evaluate refinement methods. Hu and Fu [14] presented a first evaluation of policy refinement methods, providing an overview of the current refinement methods. However, their approach does not consider sustainability-oriented policies.

The remainder of this article is organized as follows: Section II presents the related background and the policy abstraction levels, based on academic literature. Section III describes what is a sustainability-oriented policy. Section IV defines the requirements that a policy refinement process should address and the extended capabilities necessary to refine sustainability-oriented policies. Section V describes and evaluates the existing policy refinement approaches considering the sustainability-oriented policy requirements. Section VI discusses the presented analysis. Final considerations are given in Section VII.

II. Background on Policies and Policy Refinement

Moffett and Sloman [12] define a policy as "a plan of an organization to achieve its objectives." Policies define the desired behavior of systems and their components. Westerinen *et al.* [15] state that a policy can be defined from two perspectives: as a goal to guide decisions, executed within a particular context; or as a set of rules to manage and control access to network resources, as defined in [16]. Strassner [17] defines a policy as a set of rules or goals used to manage or control access to a set of resources and services of Information and Communication Technologies (ICT).

Most authors divide policies in two types: one to state what must be done in the system, given a certain condition, called obligation or management policy; and another to define what is allowed or not, restricting the system access, called authorization or access control policy [17], [18]. Authorization policies are expected to be less dynamic than obligation policies, since the latter are triggered by events, but are dormant until the event occurs, while authorization policies are acting all the time [18]. Wies [19] extended this classification including information regarding lifetime, geographical scope, organizational structure, type of service, targets, and management functionalities to which the policy applies. The author then uses this criteria list to derive attributes for policy templates.

A. Policy-Based Management and Policy-Based Network Management

Considering the challenges and the complexity in the management of large distributed systems, especially sustainability-oriented ones, the management of systems driven by policies is being widely used to decrease the management complexity. This is called Policy-Based Management (PBM) and in the case of networks, Policy-Based Network Management (PBNM).

Strassner [17] defines PBM as the application of policy rules to manage the configuration and behavior of entities. Agrawal *et al.* [20] stated that the aim of PBM is to operate and dynamically reconfigure computing resources guided by rules, so that the system goals can be achieved and the system can react in a more agile way. Boutaba and Aib [10] cite the maintenance costs, flexibility, and adaptability as PBM advantages. The same authors [10] presented an extensive work on the historical perspective of policy-based management until 2007. Fig. 1 presents the topics discussed and some examples.

Besides presenting the different types of policies, the authors present the history of works related to security, management, and networking policies, such as the framework proposed by the Internet Engineering Task Force (IETF). The framework represents the basic components a policy system must consider: a management tool, a human friendly interface for policies specification; a policy repository; a policy decision point (PDP), which controls the system and decides the actions that are going to be enforced; and some policy enforcement points (PEP), which will apply the decision taken by the PDP [21], as illustrated in Fig. 2. IETF also proposed the Common Open Policy Service (COPS) protocol to exchange information between the PDP and the PEP [22].

Policies are created, modified and stored by the Policy Management Tools and the Policy Repository; searched and retrieved by the PDP; and enforced by the PEP. This architecture presents the essential components that a PBNM system

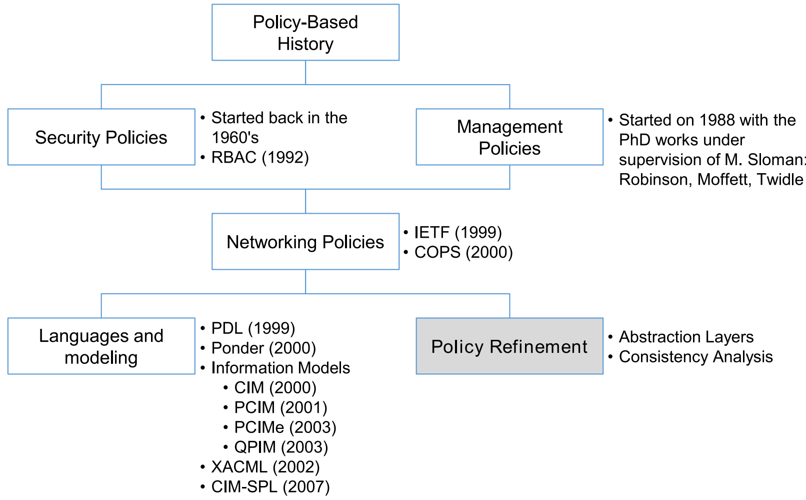

Fig. 1. Policy-based management relationships and topics based on [10].

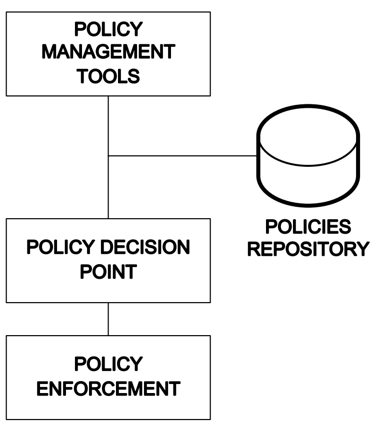

Fig. 2. IETF architecture for networking policies [21].

must comprise, in a device and vendor independent manner, interoperable and scalable [21].

More recently, with the cloud infrastructures development, policies have been gaining more importance. The infrastructures are getting bigger, the need for automation is gaining importance, and so does the policy-based management in such environment. There are different inputs driving policies, such as regulations, privacy concerns, application requirements, and business rules [23].

For OpenStack environments, for instance, some of the recent efforts are in Congress, a "project to provide policy as a service across any collection of cloud services in order to offer governance and compliance for dynamic infrastructures." It ensures that applications managed by the orchestration module (in the case of OpenStack, the Heat module) are consistent with business policies across different resources, such as compute, storage, and network. It works in conjunction with isolated policy engines, such as the Neutron for networks in OpenStack environments, being a single point of entry for administrators to define policies that are later distributed for the enforcement points [24].

B. Policy Representations

Several proposals have been made to represent policies. The Policy Description Language (PDL) [25], developed in 1999 at Bell-Labs, is an event-based language. It has a simple syntax that represents event-condition-action rules, only supporting obligation rules [26]. Ponder [27] is the most widely-deployed policy language [28]. The language name became associated with the complete toolkit to specify, analyze and enforce the policies. The last version, Ponder2, is a language for security and management policies, based on the object-oriented paradigm [29].

IETF proposes using an object oriented information model to represent policies. The Policy Core Information Model (PCIM), later extended to PCIMe [16], defines policies rules and their different parts in a vendor independent manner, supporting the definition of different levels of abstraction. It was based on the Core Information Model (CIM), a conceptual framework for the schema of the managed environment [30]. The Quality of Service (QoS) Policy Information Model (QPIM) [31] specializes PCIM to deal with QoS management. In QPIM and PCIMe, all components of a policy are represented as classes. Representing policies using information models has an advantage: the classes can be mapped to structure specifications, such as XML. These structures can then be implemented in the policies repository [26]. Fig. 3 exemplifies what an information model is with a small excerpt of a bigger information model.

C. Policy Abstraction Levels and Policy Refinement

Regarding policy refinement aspects, the focus of this survey, the first important point to consider is that a PBNM solution must offer a particular level of abstraction to the network administrator, expressing policies in a high-level language rather than as sets of configuration parameters or commands specific to particular types of network devices [14].

A policy can be seen within a hierarchy of different abstraction levels, according to the way it is expressed, communicated,

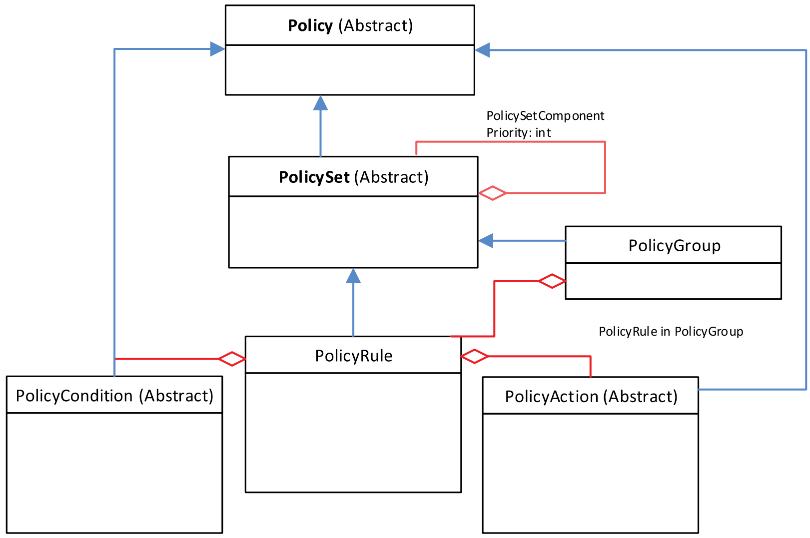

Fig. 3. Information model example.

TABLE I
POLICY LEVELS PROPOSED BY DIFFERENT AUTHORS

Policy Type	Maullo and Callo [32]	Sloman [18]	Wies [19]	Koch [33]	Strassner [17]
High Level	Societal (principles)	Goals (actions or operations that have to be interpreted by humans or refined by an expert, application dependent systems)	Corporate or high-level (derived from corporate goals)	Requirements enterprise viewpoint)	Business (SLA, processes, guides and goals)
	Directional (goals, such as organizational and corporate goals)				
	Organizational (practices)		Task oriented policies (related to process management)	Goal (information viewpoint)	System (details the business, including metrics, device and technology independent policies
	Functional (targets, policy maps to more precise methods like configuration specifications or workload targets)				
Interm. Level	Process (guidelines, in some structured language)	Rules (actions or operations that can be executed by automated tools)	Functional (define the use of management functions)	Operational (computational viewpoint)	Network (structures, languages, device independent, technology specific policies)
					Device (device and technology specific)
Low level	Procedure (rules, encoded procedures that are executables)	Mechanism information, rules for implementation	Low level (operate at the level of managed objects)		Instance (device specific commands)

or automated [32]. There may be several layers of policies, starting at high-level business policies, down to rules for implementation in network devices [18]. The number of layers may be arbitrary and application specific. Koch *et al.* [33] state that the minimum number of layers is three in order to bridge the gap between the highest and the lowest level. Strassner [17] proposes the *Policy Continuum*, composed by five levels of abstraction: Business, System, Network, Device, and Instance Levels.

Table I summarizes different levels proposed by different authors using three layers as the basis for classification: high-level, in which the business goals are expressed; intermediate-level, in which the policies are more structured; low-level, in which there are procedures and implementation mechanisms. The views vary from high-level device-independent to low-level device-specific, thus enabling different constituencies to detail policies with a proper terminology [34].

The translation process from high-level to low-level is referred to as *Policy Refinement* [32]. Policy refinement is the process of translating a policy described in a high-level of abstraction (business rules, operator language) into the device-specific corresponding configuration [17], [35]. Bandara *et al.* [35] provide a more formal definition of policy refinement:

If there exists a set of policies $P_{set} = p1, p2, ..pn$, such that the enforcement of a combination of these policies results in a system behaving in an identical manner to a system that is enforcing some base policy P_{base}, it can be said that P_{set} is a refinement of P_{base}. The set of policies $P_{set} = p1, p2, ..pn$ is referred to as the refined policy set.

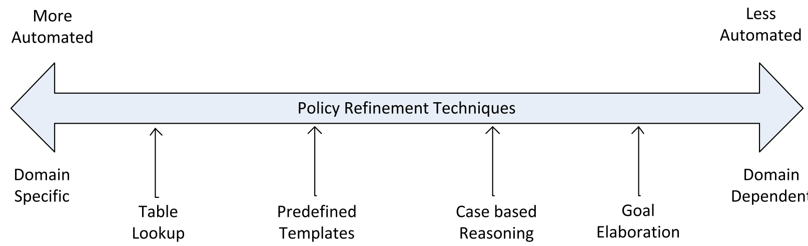

Fig. 4. Automation level of policy refinement techniques.

According to Craven *et al.* [36], the automation of policy refinement promised important benefits, but very few concrete approaches have emerged since then. Lobo *et al.* [37] stated that the policy refinement should be as automated as possible because the manual process is error prone and very dependent on a specialist, thus potentially incurring high costs. Verma [38] supports this idea by claiming that, with the increasing technical complexity, it is becoming difficult to find trained personnel who can manage new features (as in our case the energy efficient ones), so that a largely automated policy-based network management would simplify the administration process. However, Bandara [35] identified a trade-off between automation and generality of an approach, showing that, the more automated the refinement, the more domain-specific it is, as illustrated in Fig. 4. This indicates that an automated refinement solution for the sustainability domain should be specific.

III. Sustainability and Policy-Based Management

Several proposals have been developed on energy efficiency capabilities, ranging from local chip-level features for more power-efficient nodes to routing protocols. Bolla *et al.* [1] classified the existing solutions to reduce energy consumption in communication networks into three categories: (1) re-engineering, which addresses the design and materials used in equipment; (2) dynamic adaptation, which deals with adapting the network according to traffic or service requirements; and (3) sleeping/standby, which puts to sleep unused devices or parts of the device. Some capabilities could be applied at the network level, thus requiring knowledge of the entire network, or at the device level, requiring only local knowledge.

Other classification of techniques and mechanisms can also be found in [9], [39], and [40]: traffic/resource consolidation, selective connectedness, proportional computing, and virtualization. Traffic/resource consolidation creates opportunities to save energy based on the network behavior (workloads), adapting the network in order to change the state of unused equipment. Selective connectedness refers to the dynamic adaptation of devices. It allows parts of the device to go idle for some time, as transparently as possible, moving network-related traffic processing from high-consumption main board CPUs to low-power devices or external proxy devices. This technique is also referred to as interface proxying in Bolla *et al.* [1] and Bianzino *et al.* [39]. Proportional computing was first introduced by Barroso and Holzle [41] and refers to the idea of the system consuming energy in proportion to its utilization, including techniques of link rate adaption. Virtualization allows more than one service to operate on the same piece of hardware, thus improving the equipment utilization.

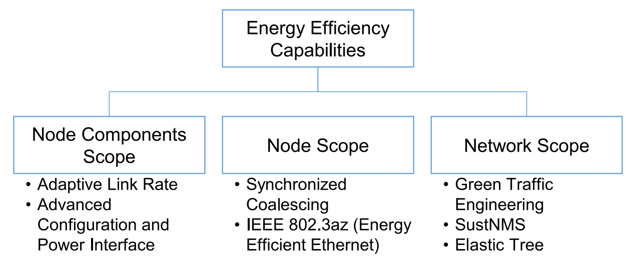

Fig. 5. Energy efficiency capabilities examples.

Schlenk *et al.* [42] proposed a taxonomy that, rather than just describing the capability (rate adaption, sleeping, and energy-aware routing), takes the scope of the capability into account: Node Components Scope (node components, memory), Node Scope (network nodes), and Network Scope, as illustrated in Fig. 5. The broader the scope, the more complex the control and more configuration options are available.

Related to the Node Components Scope, Adaptive Link Rate (ALR) allows to reduce or increase the link rate between two interfaces in accordance with the traffic, using existing Ethernet data rates [43]. Advanced Configuration and Power Interface (ACPI) performs rate adaption and sleeping in parts of devices [44]. In the Node Scope, Synchronized Coalescing (SC) [45] uses traffic bursts to create idle periods, during which it is possible to put devices to sleep. While the device is sleeping, incoming packets are buffered until a threshold is exceeded. The device then wakes up and forwards the buffered packets. IEEE 802.3az (Energy Efficient Ethernet) defines mechanisms to put interfaces in idle mode when there is no data to send, in a way that allows it to wake up quickly when a new packet arrives [46]. Related to the Network Scope, Green Traffic Engineering (GreenTE) [47] is used to free some links by moving their traffic onto other links. SustNMS, a sustainability-oriented policy-based network management system [7], was designed inspired on GreenTE and network-level policies to analyze trade-offs between energy efficiency and reliability. With information from power profiles of devices, it performs green traffic engineering and puts unused devices to sleep. ElasticTree [48] was designed for SDNs and uses the similar approach of concentrating traffic to put unused devices to sleep, focusing on fat-tree topologies in datacenters. These capabilities have different parameters that can be configured according to the necessity. Examples are, in SC, the DutyCycle (the amount of time the equipment must remain in sleep mode), tOn (the time the equipment must remain in operation mode), and threshold (the number of packets that determine if the node should remain operating or sleeping), and, in ElasticTree, the safety margin (the higher the safety margin, the higher the reserved bandwidth and the possible performance).

A. *Sustainability-Oriented Policies*

Policies have been commonly used to specify Quality of Service (QoS) and access control rules [49]. Now, with the new requirement of making networks more energy efficient, policies should encompass sustainability. We define a sustainability policy as follows:

A sustainability-oriented policy is a policy that manages energy efficiency features in the network.

Relaxing QoS requirements in ICT systems may yield opportunities to achieve more energy savings [50]. According to the authors, a "Green SLA" is a type of Service Level Agreement (SLA) that offers an extended scope of energy optimization by relaxing the existing performance parameters, introducing new energy performance parameters, and providing incentives to the customers in exchange for a specified performance degradation of the services.

B. *Abstraction Levels of Sustainability-Oriented Policies*

To complete the definition of sustainability-oriented policies, we present definitions and an example for each abstraction level defined in Carvalho *et al.* [11]. The authors propose a methodological approach for sustainability-oriented policies refinement that used the Policy Continuum levels defined in [17] as basis.

A **Business Level** policy expresses business goals or service-level agreements (SLAs) parameters, defined in the Service Level Specifications (SLSs), the technical specifications deriving from the SLAs. Goals can be established inside a company, as the operational policies described in [51], to accomplish sustainability objectives, such as to reduce energy consumption or greenhouse gas emissions. Goals can also be established based on SLAs. For sustainability, a promising approach is to allow a relaxation of traditional SLAs as proposed by [50]. Furthermore, the business level policies also need to encompass the description of objectives in terms of energy efficiency.

The **System Level** describes the operation of a policy in a device and technology independent fashion, without using networking specific terminology. At this level, it is expected to specify the metrics to accomplish the SLAs and the goals. Table II clarifies the possible metrics that are consolidated at this level and also exemplifies sustainability metrics in different abstraction levels.

A **Network Level** policy is device independent, but technology specific. It is commonly expressed in an Event-Condition-Action form. Sustainability-oriented policies in the network level should comprise obligation policies, e.g., to put the routers in a given path to sleep if the workload is smaller than 20%, and authorization policies, e.g., to allow a network function to put the router to sleep. This level faces a greater degree

TABLE II
EXAMPLES OF SUSTAINABILITY METRICS IN THE POLICY CONTINUUM

	Metric	Details
Business	Energy related OPEX	Operational expenditures with energy
System	Watts-hour per energy source	Amount of energy used per different energy source
	Power Usage Effectiveness (PUE)	Total Facility Power divided by IT Equipment Power [52]
Network	Network Energy Consumption Rating (NECR)	Watts/bits of a set of equipment [53]
	CO_2 emissions	Given the energy consumption of the network and energy source [54]
Device and Instance	Energy Consumption Rating (ECR)	Energy Consunsumption divided by the Effective System Capacity [52]
	EPI (Energy Proportionality Index)	(Power Consumption at Maximum Workload (PM) - Power Consumption at Idle Mode (PI)) divided by PM multiplied by 100% [52]

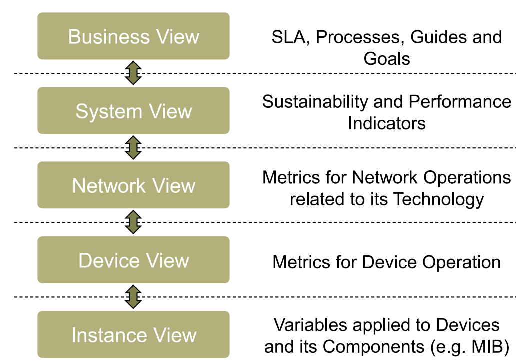

Fig. 6. Sustainability-oriented policy refinement methodological approach [11].

of dynamicity than the system level. For instance, suppose a virtual router that was migrated to a new locality. Although such an operation is transparent to the system level, the network level must take into account some particularities of the new locality, such as another energy source or another underlying topology.

The **Device Level** is device and technology specific, implying that a policy at this level is described with respect to protocols and features directly supported by a network node. The role of the devices is also relevant at this level. An administrator can thus create a green role for devices that take part in a green solution. An example of a policy at this level would be "*Enable ALR and set the link rate to 10 Mbps on the Interfaces*."

The last level of the policy continuum is the *Instance Level*. The instance level policies express the machine-readable commands (e.g., NETCONF, SNMP, OpenFlow, or CLI commands) for each device. This level is tightly related to node and vendor-specific characteristics and to particular software releases. An example of a policy at this level would be "*netconf enable ALR maxRate* $= 10Mbps$." Fig. 6 summarizes all the levels for sustainability-oriented policies.

Section IV presents the requirements a method should fulfill in order to refine the policies between these levels.

IV. Sustainability-Oriented Policy Refinement Requirements

After defining energy efficiency capabilities and sustainability-oriented policies, in this section we exploit the requirements a refinement method should fulfill in order to support the refinement of policies in general, including the support to sustainability-oriented policies.

As the requirement (i), the refinement process should comprise translation steps (automated or human guided) [12]. A policy refinement approach considered fully automated must refine the policies from the highest level down to the lowest level of abstraction, where the actions are applied. This translation could interpret the semantics of policies or could be limited to a syntactical transformation which, albeit more limited, could meet the refinement requirements [19].

As requirement (ii), the transformation process should take into account the resources presented in the network [19], [38]. This includes any type of related resources, such as the equipment and the capabilities available (such as QoS or energy efficiency capabilities).

The requirement (iii) is the verification if the refined policies meet the requirements of the original policy [12]. This relates to coverage analysis, which verifies if the refined policies cover all the high-level objectives. Examples of coverage analysis are to check if every member of the high-level policy is addressed by the lower-level policies, or if at least one authorization policy is applied to each object under management. There are much more challenging problems, and the completeness is hard to achieve [18].

Coverage is part of a broader study area called Policy Analysis, which deals with coverage gaps, policy comparison, behavioral simulation, and conflicts. A conflict may occur when one policy interferes in the behavior of another policy, denying its action or putting the managed objects in undesired states. This happens when there is an overlap between subjects or targets. The task of detecting (and solving) conflicts, the requirement (iv), is extremely difficult and is generally related to authorization policies. For other types of policy, conflict detection demands application knowledge or some human intervention [18], [55]. Verma [56] stated that, if a policy is represented without any constraints, the policy conflict detection can be shown to be NP-complete. Therefore, some constraints are required when representing policies. Considering these constraints, the author proposed a solution using topological spaces.

A simpler solution to detect conflicts is to search for overlaps in each pair of policies. Such solution has a running time of $O(dimensions * number\_of\_policies^2)$ [56]. The author suggests combining these and other simplification techniques to improve the running time of detecting conflicts. And, to solve the detected conflicts, the most common way is to make the administrator choose which policy has more priority, the approach also used by Kagal [57]. The conflicts can be solved during translation, starting at the highest level, or in a lower level. Solving conflicts during translation results in a more complex process and may cause problems in dynamic domains [19].

Conflicts detection and resolution is a requirement of any type of policy. However, there can be more conflicts in environments where sustainability-oriented policies are presented. This is due to their antagonic characteristic: whereas the sustainability-oriented policies work to reduce energy consumption, the usual QoS policy tries to maximize performance, thus maybe not enabling energy savings. Relaxing QoS requirements may enable opportunities to achieve more energy savings [58].

Policy refinement methods should also deal with dynamicity, the requirement (v). That is, apply policies in different time slots or be able to determine what to do when the scenario changes (for instance, when a node migrates to another network). To deal with temporal dynamicity, Sloman [18] cites policy constraints, which are predicates referring to global attributes such as time or action parameters. They can define allowed values in management operations, or define preconditions. Regarding scenario changes in an energy efficiency case, the policies could express the migration of processes between machines when the workload achieves a particular value and the system is not in peak hours, for example.

Wies [19] suggests treating scenario changes like the initial enforcement actions. This is because changing a target or an action, for instance, may require a complete new refinement. The same author complements the idea by stating that a policy should be able to emit notifications when any of its parts changes, so that the necessary actions can be performed. Monsanto *et al.* [59] propose to use parameterized policies. The parameters of the policies can be updated whenever a scenario changes.

Considering sustainability-oriented policies, dynamicity gains importance, given the attempts to save energy that take advantage of time periods in which the bandwidth utilization is low. This way the network administrator can take advantage of the significant difference between the usage rates during day and night.

Dinamicity is also paramount in strategies that attempt to move, for instance, a virtual node to a more energy efficient location, or to a location with a different energy matrix. This would imply changing the parameters of the policies of this virtual node, which now may be in a location with different probe rate, power consumption, or even green capabilities.

As the next requirement, (vi), the method should be able to represent policies in order to keep context, coherence, and integrity of the network under determined conditions [32]. This implies that, in order to handle sustainability issues, besides supporting traditional obligation and authorization policies, the method should be flexible to accept sustainability metrics, energy efficiency rules, and interface with new features, such as changing the equipment chip's operating frequency to save energy or put a router to sleep. Without modeling, for instance, the sleeping action, the system would not be able to enforce the action during the operation. Or, without modeling a new type of variable, such as the Watts/bits ratio, the system will not be able to monitor this value and take the necessary decisions.

Maullo and Callo [32] suggested using object-oriented modeling for such system representation. PCIM (and PCIMe) represents policies using classes. An information model using object-oriented diagrams, such as UML, could be developed

TABLE III
REQUIREMENTS SUMMARY

Requirement	Summary
(i) Translation	Refine down high-level policies, considering the different abstraction levels
(ii) Resources	Take into account the resources in the underlying network, including the capabilities available
(iii) Verification/ Coverage	Verify if the refined policies fulfill the requirements of the original, high-level policy
(iv) Conflicts	Detect and solve conflicts among policies
(v) Dynamicity	Deal with dynamicity of time and scenario changing
(vi) Sustainability representation	Represent sustainability-oriented and other types of policies, including metrics and specific actions
(vii) Capabilities orchestration	Orchestrate (coordinate and combine) energy efficiency capabilities to save more energy and ensure a conflict-free operation

to represent sustainability-oriented policies and all the components and relationships among them. Such information model can be an extension of PCIMe.

Another requirement, (vii), is the orchestration of green capabilities, that is, the method should be able to (a) choose the best capability considering the network situation, (b) combine different capabilities in order to increase the energy efficiency, and (c) avoid combining conflicting capabilities, such as simultaneously applying on the same node two capabilities that put the device to sleep. In this case, there could be a conflicting situation in which the capabilities act in opposite ways, leading to savings negation or even failures.

Table III summarizes the sustainability-oriented policies refinement requirements. These requirements model the desired behavior of a refinement method which, besides supporting the refinement of different types of policies, also enables energy efficiency capabilities orchestration, a mandatory requirement considering the current efforts on saving energy. The requirements and the trade-off between generality and automation presented in Section II show that it is not an easy task to refine sustainability-oriented policies through the different policy abstraction levels. For example, it involves the translation of sustainability-related parameters and metrics from high to low-level, the construction of new network rules, and the binding of network rules with specific green technologies.

In a large distributed system, the management of policies in general, not only of sustainable ones, is a very complex and error-prone task, usually requiring experienced professionals. A policy refinement method for sustainability-oriented policies should be conceived fulfilling the discussed requirements and minimizing the human dependency and intervention in the network management, thus decreasing the probability of errors and the costs involved in retaining trained professionals specialized in low-level tasks.

Section V evaluates different existing approaches to policy refinement, with respect to the discussed requirements. The evaluation focuses on the level of automation of the approaches and on how easily they can be adapted to refine sustainability-oriented policies.

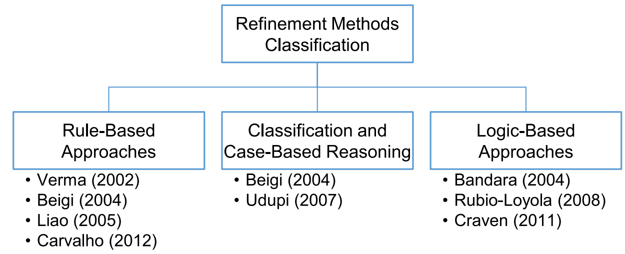

Fig. 7. Refinement methods classification.

V. ANALYSIS OF THE POLICY REFINEMENT APPROACHES

In order to evaluate the existing approaches to policy refinement, we take into consideration the requirements listed in Section IV and the classification of methods we present in Fig. 7. This figure presents our proposal for grouping the methods according to their approaches and evolution in time. The first category, Rule-Based Transformation, groups the approaches that propose pre-defined rules to perform the refinement. Such approach tend to be more domain specific and more automated. The second category, Classification-Based Refinement and Case-Based Reasoning, relates to those approaches that use some learning techniques to perform the refinement. Albeit relating to just one abstraction level translation, they represent important techniques that could be used to improve a complete refinement method. The third category, Logic-Based Approaches, groups more generic methods that use more formal representation of policies. They tend to be more generic and less automated in nature, demanding some effort on modeling the application so that the method can be automated.

A. *Rule-Based Transformation*

The approach of Verma [56] is domain specific and one of the most automated approaches. It uses table lookup techniques for network QoS management. It defines only two levels of policies: one at the business level, another at a technology-specific level, between which the refinement method is automated. The author proposes to use tables to relate users, applications, and devices to classes of service. The method performs table lookups to build the relationships during the refinement, thus depending on the correctness of the table contents. This drawback is compensated by the easiness of analyzing contradictions and coverage of such a rule-based notation.

The author proposes a module able to determine the network topology, users, and applications, in conjunction with the capabilities available. The validation of results is certified by the table lookup approach, responsible for validating the information and bounds of policy attributes and parameters. Regarding policy analysis, the author mentions coverage verification and propose algorithms to be used to check policy conflicts and unreachable policies. To detect conflicts, the author suggests using topological spaces. Any potential conflict detected is solved through the attribution of priorities to the conflicting policies. The author also checks feasibility, that is, if the refinement target can be achieved. This can be determined by using queue models to predict if the policy target is going to be achieved.

TABLE IV
RULE-BASED APPROACHES SUMMARY

Approach	Summary
Verma [56]	Table Lookup, domain specific, automated approach
Liao and Gao [61]	Extended [56], proposing policy templates to turn the method more generic
Carvalho et al. [11]	Based on [56], but methodological, focusing on sustainability-oriented policies

The approach models the period of the day in the policies, but does not model scenario changes. As the method is only able to handle QoS policies, it does not support the representation of sustainability-oriented policies and the orchestration of green capabilities. Beigi *et al.* [60] classify this approach as part of the Static Transformation category of their tripartite classification.

Liao *et al.* [61] extend the translation of Verma [56] by proposing a not domain specific approach based on recipes, i.e., policy templates. The recipes define all possible refinement alternatives for each business level policy, which are branches describing possible steps based on high-level policies. The policy refinement engine automatically refines policies by choosing the refinement template, based on the conditions of the templates. The refinement process starts with the policy refinement engine receiving a tagged abstract policy and recipes. If there is a match between the tags and a policy in the repository, the refinement engine produces a concrete policy (i.e., ready to be applied) or an enforceable policy (i.e., to be used by an agent).

Carvalho *et al.* [11] propose a methodological approach for sustainability-oriented policies refinement, from the business level down to the instance level, with grounds on the rule-based methods and based on the Policy Continuum described in [17]. The process of policy refinement starts at the business level, where high-level policies are translated down to the system level by incorporating sustainability and performance indicators. The system level policy is operationalized at the network level using the Ponder2 framework [62]. The policies described in Ponder2 must be interpretable by the devices in the network. The device level uses a protocol, such as Simple Network Management Protocol (SNMP), to refine the policy to the instance level. The instance level policy then applies actions and provides information for the upper layers. This approach can be seen as a first step towards an automated policy refinement for sustainability-oriented policies, defining a methodology that could be used to develop an automated approach to the policy refinement problem. Table IV summarizes the presented rule-based approaches.

Regarding the refinement requirements, this category addresses the automated translation, resources required, verification and coverage analysis, and conflicts detection and resolution requirements. The dynamicity requirement is addressed partially, since time conditions can be modeled, but the dynamic scenario can not. Regarding sustainability representation, Carvalho *et al.* [11] supports it, and the other approaches could be adapted to accept sustainability information models. To conclude, regarding the seventh requirement, orchestration, the approaches that fall in this category do not address it. The proposed approaches deal with only one management capability, not being able to handle different elements and capabilities.

B. Classification-Based Refinement and Case-Based Reasoning

Classification-Based Refinement and Case-Based Reasoning (CBR) approaches usually have just two abstraction levels, including the implementation level. Nonetheless, they represent important techniques in the policy refinement field. Beigi *et al.* [60] detail three approaches to perform refinement: using static rules, table lookup, and applying case-based reasoning, which is more detailed. The case-based reasoning approach uses the knowledge learned from the behavior of the system in the past to predict its present and future behavior. The system maintains a database of previous cases, in which each case is a combination of business objectives and configuration parameters corresponding to that goal. When a new configuration is needed, the system tries to find the closest matching case in the database, or an interpolation between a set of matching cases to determine the appropriated configuration. However, the authors affirm that the effectiveness of the approach depends on having a rich enough set of cases to be consulted in the database.

According to Boutaba and Aib [10], the case-based reasoning approach has a number of weaknesses, such as the difficulty to populate the case database, and the possibility of false acceptance due to "generalizations made based on wrongly constructed sets of cases." The main advantage would be that the system "becomes increasingly effective as its case database grows in size." The authors also state that a policy transformation mechanism should be used in conjunction with CBR. The work of Beigi *et al.* [60] could be used in conjunction with other modules to fulfill sustainability-oriented requirements, in addition to providing classification and learning features.

The approach of Udupi *et al.* [63] statistically classifies relevant low-level system attributes through static rules and decision trees. This is performed in order to generate policies to maintain the relevant attributes for system health. To execute this, the approach counts with four main phases:

1) Test and Development: in this step, the method creates a specific system configuration on the given high-level policies and executes a workload. This workload is a manual input of data around the target high-level policies to generate specific results in order to perform the classification phase;
2) Classification: in this phase, a classification is performed on the set of data collected before, based on a classification algorithm to generate a decision tree. The decision tree is used to verify if a path satisfies or not the high-level policies;
3) Policy Derivation and Refinement: this phase derives the paths generated in the classification phase into policies. The refinement strategy applied at this level uses the distribution statistics of the attributes on these true paths;
4) Allowed and Restricted Ranges: the parameters that allowed ranges may have are derived from all the refined policies, by union operations over the individual allowed ranges.

TABLE V
CLASSIFICATION AND CASE-BASED REASONING APPROACHES SUMMARY

Approach	Summary
Beigi et al. [60]	Case database of previous cases (combination of business objectives and configuration parameters). Increasing effective as the database grows
Udupi et al. [63]	Classification of parameters to generate policies with relevant attributes for the system

The test and development phase relies on a manual input of workloads and static rules and the main goal is to classify important parameters. Besides, the authors state that their method was developed for performance related goals, but the method could be extended to comprise sustainability-oriented goals. The approach presents a useful method for system monitoring and health check when it is deployed and running. Table V summarizes the presented classification and case-based reasoning approaches.

Regarding the refinement requirements, this category addresses partially the automated translation, since it deals with only one level up to the implementation. The methods do not cover the resources required, verification and coverage analysis, and conflicts detection and resolution requirements either. The dynamicity requirement is addressed partially, since time conditions can be modeled as a parameter, but there is nothing related to the dynamic scenario. Regarding sustainability representation, the methods could model sustainability parameters. To conclude, regarding orchestration, the category does not deal with capabilities, but with parameters and metrics.

C. Logic-Based Approaches

Bandara *et al.* [35] proposed a not automated method having Event Calculus as the base formalism. The first step is to translate abstract goals into operationalized goals, relying on the KAOS methodology. The second step takes these goals and maps them to specific modules. The method demands the system description with domain specific information: objects and respective domains, policy rules, and available management operations with the objects they affect. The authors suggest modeling the system with UML. The relationship between the goals and the system description is called a Strategy [10]. After describing the system and goals, the next step is to perform abductive reasoning, which allows to derive the facts that must be true for goals to be achieved considering the system description.

Rubio-Loyola *et al.* [64], instead of using abductive reasoning, applied model checking to derive low-level actions from high-level policies. The authors presented a policy refinement framework, applicable to any domain, grounded in goal-elaboration methodologies and analysis of reactive systems. The approach uses Linear Temporal Logic to define relationships between goals, needing one expert to define the goals, another to select which to use in each case, and an automated policy encoding to translate the defined goals into Ponder2 expressions [62].

In both cases, the goals were modeled in a formal manner, what can help, for instance, in verifying if the refined policies meet the high-level policies requirements. Charalambides *et al.* [55] show how such technique can be used to detect conflicts that emerge at run-time, besides presenting a proposal for specifying policies to automate conflict resolution. A set of rules with logical predicates detects and signals conditions where conflicts may occur. To solve the identified conflicts, the authors proposed attributing different priorities to the policies. However, such approach may not solve all conflicts, and some of them may require human intervention for resolution. Application-specific conflicts are even harder to treat since they can depend on the current state of the system. In this case, the network administrator can predefine policies that provide a resolution in case a conflict occurs.

Craven *et al.* [65] proposed an automated policy refinement method based on four stages: policy decomposition, operationalization, re-refinement, and deployment. The inputs of the approach are:

- The initial business level policy, defined in a structured natural language;
- The domain description, which is a UML model containing a representation of the structure of the classes, kinds of possible associations, possible operations on instances of the classes and an instance repository, that records the objects existing in the domain and the relations between them;
- Obligation or authorization policies that are decomposed and operationalized; and
- Decomposition rules representing how actions and objects described at high level relate to those at a lower level.

The authors use a variant of Event Calculus to describe the state of the system and to express conditions under which a policy applies. The refinement process interleaves two stages: decomposition and operationalization. The decomposition stage receives the decomposition rules given on the input and matches the operationalized policies with object classes. The operationalization stage uses the domain description and the high-level policies to provide information about how an action can be implemented.

At this stage, the resources to which actions should be applied are defined by a comparison of the instance repository with the conditional statements of the policies. Then the policies are tested to assert whether they are expressed in terms an enforcement point can understand. Finally, if necessary, the decomposition stage is performed again to guarantee the accuracy of the refined instance level policies. The decomposition rules relate actions to components. In a dynamic scenario, new decomposition rules must be defined. The authors say that, in such cases, re-refinement can be a way of automating the necessary adjustments, but do not give further details. The authors also state that a powerful policy analysis component is essential [66].

The logic-based approaches can be adapted to interpret sustainability metrics, rules, or actions, and the domain description could also encompass them. However, to address the

TABLE VI
LOGIC-BASED APPROACHES SUMMARY

Approach	Summary
Bandara et al. [35]	Event Calculus to define policies, environment defined using UML. Abductive reasoning for refinement
Rubio-Loyola [67]	Refinement using model checking instead of abductive reasoning, more automated
Craven et al. [65]	Variant of Event Calculus to define policies, domain described using UML. Two stages for refinement: decomposition and operationalization

orchestration of green capabilities, they would require the implementation of a whole new module as an extension. Table VI summarizes the presented logic-based approaches.

Regarding the refinement requirements, this category addresses the automated translation, resources required, verification and coverage analysis, and conflicts detection and resolution requirements after the domain is modeled. The dynamicity time requirement is addressed, but the dynamic scenario is not. Regarding sustainability representation, the domain modeling could comprise sustainability aspects, so that the methods could handle such parameters. Like the others, this category does not address the orchestration requirement. The proposed methods detail only one management capability being applied. It is not possible to use the evaluated methods for orchestration of various capabilities at the same time.

D. Other Related Initiatives

KAoS [68] is a framework that uses ontologies to define policies and the relationships among parameters. Agrawal *et al.* [69] define KAoS as "a collection of componentized policy and domain management services originally designed for governing software agent behavior, and then adapted to grid computing." A policy is specified in the KAoS Policy Administration Tool (KPAT) module. The policies are then translated using predefined ontologies to a format that can be monitored and enforced. The method is also able to detect conflicts, but the decision on what to do after detecting a conflict relies on the network administrator. The method is applicable to any domain, since the domain is ontologically modeled in the system [70], [71]. Automated policy refinement is mentioned by the authors, but not further detailed.

Recently, with the gaining importance of SDN, some approaches to PBNM have been proposed with focus on such environment. As examples, we describe below Procera [72] and the work of Monsanto *et al.* [59].

The authors of Procera [72] propose a controller architecture and a control language intended to offer more expressiveness in SDN domains, providing means for network operators to express policies in an easier way. It is based on principles of functional reactive programming, which consist of continuous time-varying behavior and event-driven reactivity [73]. Additionally, Procera responds to events from sources other than OpenFlow, such as events triggered by user authentication or use of bandwidth. The approach is intended to network operators, and the high-level policies must be manually translated down to the network level.

The work of Monsanto *et al.* [59] presents another policy-based approach for SDN environments. They propose a framework on top of a POX controller and use the syntax of the Pyretic language to allow high-level definition of policies. Pyretic is a platform embedded in Python language that embodies concepts such as packet-forwarding policy, network conditions monitoring, and dynamic policy to respond network events, enabling network operators to create sophisticated SDN applications. The Pyretic language extends the Frenetic project [74], a collaborative effort between researchers to develop a language for SDN applications. The method also proposes parameterized policies, similarly to what was proposed in [75]. This allows to update the parameters of the policies whenever a scenario changes. The Pyretic language provides several features in order to support network management, such as QoS support with rate limiting and prioritization, which is useful to sustainable purposes. However, like Procera, the highest level in this solution is focused on network operators, not suiting for business level.

VI. DISCUSSIONS AND FUTURE WORK DIRECTIONS

Table VII summarizes the analyzed categories of methods with respect to the attendance of the seven requirements. The table also highlights the trade-off between automation and generality. None of the evaluated methods complies with every requirement of a refinement method for sustainability-oriented policy.

A. Lessons Learned

A desirable policy refinement method should be fully-automated in order for a network management to support sustainability. From the literature survey, we conclude that the more restricted an application domain is, the more chances there are to develop a fully automated solution for refinement. To satisfy the sustainability requirements, the policy refinement method should thus be domain-specific. A generic approach demands much effort from an expert to provide application specific information. Thus, in addition to being automated, in order to be effective a method should be specific and extensible.

Among the evaluated methods, it can be verified that the most automated approaches are those related to rule-based refinement, which are more domain specific. The logic-based methods are more generic, but demand a significant effort on modeling the system, for example with UML. The approaches generally support policy translation. Rule-based and logic-based are the most complete categories and allow to fulfill the requirements of resources discovery and policy analysis, i.e., verification and conflict detection. The temporal dynamicity requirement is usually fulfilled, but scenario dynamicity needs more effort in order to be fully automated. Energy efficiency policies may demand more of this requirement considering their more dynamic behavior.

Regarding the sustainability requirements specific to representation, the methods could handle metrics, events, and actions

TABLE VII
REFINEMENT METHODS CATEGORIES COMPARISON

	Rule-Based Approaches	Classification and CBR	Logic-Based Approaches
Automation x Generality	The most automated	Somehow automated, not complete refinement	Not domain specific, less automated, demands modeling
(i) Translation	√	Partial: only one level up the implementation	√
(ii) Resources	√	×	√
(iii) Verification/ Coverage	√	×	√
(iv) Conflicts	√	×	√
(v) Dynamicity	Partial: only time	Partial: can handle time as a parameter	Partial: can handle time, but dynamic scenario is not detailed
(vi) Sustainability representation	Partial: Carvalho does, others should support information models	Partial: could model sustainability parameters	Partial: could model sustainability as an input to the method
(vii) Capabilities orchestration	×	×	×

related to energy efficiency after some adaptation. On the other hand, for the orchestration scenario, a completely new module is required. There is no method able to coordinate energy efficiency capabilities considering conflicts and determining which single capability or group thereof is the best option for a given network scenario.

B. Future Work Directions

In order to fulfill the sustainability-oriented policies refinement requirements, a new method should be developed, mainly to address the orchestration requirement, since there is no existing solution to this issue. Kephart [76], and more recently Bradshaw *et al.* [28], proposed that utility function policies, which can be seen as extensions of goal policies, are key for the future of PBM. Utility Functions may be interesting because they combine values for different parameters, expressing an optimization objective. However, they would only be really useful if associated with interfaces and algorithms in order to be more user-friendly [76].

Additionally, the area of policy-based network management could benefit from an analysis that goes beyond sustainability. For instance, studying how QoS and access control capabilities could be fully supported by a complete and automated refinement method, including the orchestration of such capabilities.

VII. FINAL CONSIDERATIONS

In this Survey, we presented an extensive background on policies and Policy-Based Network Management, the definitions, and refinement requirements for sustainability-oriented policies. The existing refinement methods were analyzed in light of these requirements. We concluded with a discussion on what was possible to learn from the surveyed literature and indicated some future work directions. As main contributions, besides defining sustainability-oriented policies, we evaluated the different proposals for policy abstraction levels, defined the relationship among the metrics and the different levels, described a complete requirements list for policy refinement, and evaluated the existing methods in light of the sustainability domain, besides classifying them in three categories. We also listed other related initiatives.

A refinement method for sustainability-oriented policies demands a complete refinement through the different levels of abstraction. Additionally, it should address other requirements, such as representation and orchestration of energy efficiency capabilities, which the existing methods do not support. The existing methods could be extended or complemented with new modules to fulfill the requirements and make the network management more automated and sustainable.

REFERENCES

[1] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, “Energy efficiency in the future Internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures,” *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 223–244, 2nd Quart. 2011.
[2] K. Bilal *et al.*, “A taxonomy and survey on green data center networks,” *Future Gener. Comput. Syst.*, vol. 36, pp. 189–208, 2014.
[3] R. Bolla, R. Bruschi, and A. Ranieri, “Performance and power consumption modeling for green COTS software router,” in *Proc. 1st COMSNETS*, 2009, pp. 1–8.
[4] K. Bilal *et al.*, “A survey on green communications using adaptive link rate,” *Cluster Comput.*, vol. 16, no. 3, pp. 575–589, Sep. 2013.
[5] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, “An energy saving routing algorithm for a green OSPF protocol,” in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.
[6] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, “An OSPF enhancement for energy saving in IP networks,” in *Proc. IEEE INFOCOM WKSHPS*, 2011, pp. 325–330.
[7] C. H. Costa, M. C. Amaral, G. C. Januario, T. C. Carvalho, and C. Meirosu, “SustNMS: Towards service oriented policy-based network management for energy-efficiency,” in *Proc. SustainIT*, 2012, pp. 1–5.
[8] R. Bolla *et al.*, “A northbound interface for power management in next generation network devices,” *IEEE Commun. Mag.*, vol. 52, no. 1, pp. 149–157, Jan. 2014.
[9] K. Bilal, S. Khan, and A. Zomaya, “Green data center networks: Challenges and opportunities,” in *Proc. 11th Int. Conf. FIT*, Dec. 2013, pp. 229–234.
[10] R. Boutaba and I. Aib, “Policy-based management: A historical perspective,” *J. Netw. Syst. Manage.*, vol. 15, no. 4, pp. 447–480, Dec. 2007.
[11] T. C. M. B. Carvalho *et al.*, “Towards sustainable networks—Energy efficiency policy from business to device instance levels,” in *ICEIS (3)*, L. A. Maciaszek, A. Cuzzocrea, and J. Cordeiro, Eds. Setúbal, Portugal: SciTePress, 2012, pp. 238–243.
[12] J. D. Moffett and M. S. Sloman, “Policy hierarchies for distributed systems management,” *IEEE J. Sel. Areas Commun.*, vol. 11, no. 9, pp. 1404–1414, Dec. 1993.
[13] T. Phan, J. Han, J.-G. Schneider, T. Ebringer, and T. Rogers, “A Survey of policy-based management approaches for service oriented systems,” in *Proc. 19th ASWEC*, 2008, pp. 392–401.
[14] J. Hu and Y. Fu, “Research on policy refinement method,” in *Proc. Int. Conf. Cyberworlds*, Sep. 2008, pp. 800–804.
[15] A. Westerinen *et al.*, “Terminology for policy-based management,” IETF, Fremont, CA, USA, RFC 3198, Nov. 2001.

[16] B. Moore, Ed. "*Policy Core Information Model (PCIM) extensions*," IETF, Fremont, CA, USA, RFC 3460, Jan. 2003.

[17] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation*, 1st ed. San Mateo, CA, USA: Morgan Kaufmann, Sep. 2003.

[18] M. Sloman, "Policy driven management for distributed systems," *J. Netw. Syst. Manage.*, vol. 2, no. 4, pp. 333–360, Dec. 1994.

[19] R. Wies, "Using a classification of management policies for policy specification and policy transformation," in *Integrated Network Management IV*, ser. IFIP—The International Federation for Information Processing, A. Sethi, Y. Raynaud, and F. Faure-Vincent, Eds. New York, NY, USA: Springer-Verlag, 1995, pp. 44–56.

[20] D. Agrawal, K.-W. Lee, and J. Lobo, "Policy-based management of networked computing systems," *IEEE Commun. Mag.*, vol. 43, no. 10, pp. 69–75, Oct. 2005.

[21] G. Waters, J. Wheeler, B. Moore, A. Westerinen, and L. Rafalow, "Draft IETF Policy—Policy Framework Architecture," Feb. 1999. [Online]. Available: http://tools.ietf.org/html/draft-ietf-policy-arch-00

[22] D. Durham *et al.*, "The COPS (Common Open Policy Service) Protocol," IETF, Fremont, CA, USA, RFC 2748, Jan. 2000.

[23] P. Balland and T. Hinrichs, "Congress: A System for Declaring, Auditing, and Enforcing Policy in Heterogeneous Cloud Environments," May 2014. [Online]. Available: https://www.openstack.org/summit/openstack-summit-atlanta-2014/session-videos/presentation/congress-a-system-for-declaring-auditing-and-enforcing-policy-in-heterogeneous-cloud-environments

[24] "Congress—OpenStack," [Online]. Available: https://wiki.openstack.org/wiki/Congress

[25] J. Lobo, R. Bhatia, and S. Naqvi, "A policy description language," in *Proc. 16th AAAI/11th IAAI*. Menlo Park, CA, USA, 1999, pp. 291–298.

[26] N. Damianou, "A policy framework for management of distributed systems," Ph.D. dissertation, Dept. Comput., Imperial College London, London, U.K., Feb. 2002.

[27] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in *Proc. Policies Distrib. Syst. Netw.*, 2001, pp. 18–38.

[28] J. M. Bradshaw, A. Uszok, and R. Montanari, *Policy-Based Governance of Complex Distributed Systems: What Past Trends Can Teach Us About Future Requirements*. Boca Raton, FL, USA: CRC, 2014, pp. 259–283.

[29] K. P. Twidle, N. Dulay, E. Lupu, and M. Sloman, "Ponder2: A policy system for autonomous pervasive environments," in *Proc. 5th ICAS*, Valencia, Spain, Apr. 2009, pp. 330–335.

[30] "CIM | DMTF Standard," [Online]. Available: http://www.dmtf.org/standards/cim

[31] Y. Snir, J. Strassner, Y. Ramberg, B. Moore, and R. Cohen, "Policy Quality of Service (QoS) Information Model," IETF, Fremont, CA, USA, RFC 3644, Nov. 2003.

[32] M. Maullo and S. Calo, "Policy management: An architecture and approach," in *Proc. IEEE 1st Int. Workshop Syst. Manage.*, Apr. 1993, pp. 13–26.

[33] T. Koch, C. Krell, and B. Kraemer, "Policy definition language for automated management of distributed systems," in *Proc. 2nd IEEE SMW*, Washington, DC, USA, 1996, pp. 55–64.

[34] S. Van der Meer *et al.*, "Autonomic networking: Prototype implementation of the policy continuum," in *Proc. 1st Int. BcN*, 2006, pp. 1–10.

[35] A. K. Bandara, E. C. Lupu, and A. Russo, "Using event calculus to formalise policy specification and analysis," in *Proc. 4th IEEE POLICY*, 2003, pp. 26–39.

[36] R. Craven, J. Lobo, E. Lupu, A. Russo, and M. Sloman, "Decomposition techniques for policy refinement," in *Proc. Int. Conf. CNSM*, Oct. 2010, pp. 72–79.

[37] J. Lobo *et al.*, "Refinement of history-based policies," in *Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning*, vol. 6565, ser. Lecture Notes in Computer Science, M. Balduccini, and T. Son, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 280–299.

[38] D. Verma, "Simplifying network administration using policy-based management," *IEEE Netw.*, vol. 16, no. 2, pp. 20–26, Mar./Apr. 2002.

[39] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A Survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart. 2012.

[40] S. K. Garg and R. Buyya, "Green cloud computing and environmental sustainability," in *Harnessing Green IT: Principles and Practices*. Hoboken, NJ, USA: Wiley, 2012, pp. 315–340.

[41] L. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[42] R. Schlenk, C. Lange, H. Lehmann, and R. Vleugel, "Taxonomy of dynamic power saving techniques in fixed broadband networks," in *Proc. 14th ITG Symp. Photon. Netw.*, May 2013, pp. 1–8.

[43] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with Adaptive Link Rate (ALR)," *IEEE Trans. Comput.*, vol. 57, no. 4, pp. 448–461, Apr. 2008.

[44] R. Bolla, R. Bruschi, and F. Davoli, "Energy-aware resource adaptation for next-generation network equipment," in *Proc. IEEE GLOBECOM*, Nov. 2009, pp. 1–6.

[45] M. Mostowfi and K. Christensen, "Saving energy in LAN switches: New methods of packet coalescing for energy efficient ethernet," in *Proc. IGCC*, 2011, pp. 1–8.

[46] K. Christensen *et al.*, "IEEE 802.3az: The road to energy efficient ethernet," *IEEE Commun. Mag.*, vol. 48, no. 11, pp. 50–56, Nov. 2010.

[47] M. Zhang, C. Yi, B. Liu, and B. Zhang, "GreenTE: Power-aware traffic engineering," in *Proc. 18th IEEE ICNP*, 2010, pp. 21–30.

[48] B. Heller *et al.*, "ElasticTree: Saving energy in data center networks," in *Proc. NSDI*, vol. 3, 2010, pp. 19–21.

[49] L. Lymberopoulos, E. Lupu, and M. Sloman, "An adaptive policy based framework for network services management," *J. Netw. Syst. Manage.*, vol. 11, no. 3, pp. 277–303, Sep. 2003.

[50] S. Klingert, T. Schulze, and C. Bunse, "GreenSLAs for the energy-efficient management of data centres," in *Proc. 2nd Int. Conf. e-Energy*, New York, NY, USA, 2011, pp. 21–30.

[51] M. Perry and M. Bauer, "Policies, rules and their engines: What do they mean for SLAs?" in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 3213, ser. Lecture Notes in Computer Science, M. G. Negoita, R. J. Howlett, and L. C. Jain, Eds. Berlin, Germany: Springer-Verlag, Jan. 2004, pp. 1164–1170.

[52] X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon, "A survey of green mobile networks: Opportunities and challenges," *Mob. Netw. Appl.*, vol. 17, no. 1, pp. 4–20, Feb. 2012.

[53] V. Manral, "Benchmarking Power Usage of Networking Devices," Network Working Group, IETF, Fremont, CA, USA, Tech. Rep., May 2010.

[54] World Resources Institute, Washington, DC, USA, "The GHG Protocol for Project Accounting." [Online]. Available: http://ghgprotocol.org/files/ghgp/ghg_project_accounting.pdf

[55] M. Charalambides *et al.*, "Dynamic policy analysis and conflict resolution for diffserv quality of service management," in *Proc. 10th IEEE/IFIP NOMS*, Apr. 2006, pp. 294–304.

[56] D. C. Verma, *Policy-Based Networking: Architecture and Algorithms*. Indianapolis, IN, USA: New Riders Publ., 2000.

[57] L. Kagal, "Rei: A policy language for the me-centric project," HP Labs, Palo Alto, CA, USA, Tech. Rep., Sep. 2002.

[58] G. C. Januario *et al.*, "Evaluation of a policy-based network management system for energy-efficiency," in *Proc. IFIP/IEEE Int. Symp. IM*, 2013, pp. 596–602.

[59] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, "Composing software-defined networks," in *Proc. 10th USENIX Conf. nsdi*, Berkeley, CA, USA, 2013, pp. 1–14.

[60] M. S. Beigi, S. Calo, and D. Verma, "Policy transformation techniques in policy-based systems management," in *Proc. 5th IEEE Int. Workshop POLICY*, 2004, pp. 13–22.

[61] B.-S. Liao and J. Gao, "An automatic policy refinement mechanism for policy-driven grid service systems," in *Proc. GCC*, 2005, pp. 166–171.

[62] K. Twidle, E. Lupu, N. Dulay, and M. Sloman, "Ponder2-a policy environment for autonomous pervasive systems," in *Proc. IEEE Workshop POLICY*, 2008, pp. 245–246.

[63] Y. B. Udupi, A. Sahai, and S. Singhal, "A classification-based approach to policy refinement," in *Proc. 10th IFIP/IEEE Int. Symp. IM*, 2007, pp. 785–788.

[64] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, and G. Pavlou, "A methodological approach toward the refinement problem in policy-based management systems," *IEEE Commun. Mag.*, vol. 44, no. 10, pp. 60–68, Oct. 2006.

[65] R. Craven, J. Lobo, E. Lupu, A. Russo, and M. Sloman, "Policy refinement: Decomposition and operationalization for dynamic domains," in *Proc. 7th Int. CNSM*, 2011, pp. 1–9.

[66] R. Craven *et al.*, "Expressive policy analysis with enhanced system dynamicity," in *Proc. 4th ASIACCS*, New York, NY, USA, 2009, pp. 239–250.

[67] J. Rubio-Loyola, *Towards the Policy Refinement Problem in Policy-Based Management Systems: A Synthesis Study*. Saarbrücken, Germany: VDM, 2008.

[68] A. Uszok *et al.*, “New developments in ontology-based policy management: Increasing the practicality and comprehensiveness of KAoS,” in *Proc. IEEE POLICY*, 2008, pp. 145–152.
[69] D. Agrawal, S. Calo, J. Giles, K.-W. Lee, and D. Verma, “Policy management for networked systems and applications,” in *Proc. 9th IFIP/IEEE Int. Symp. IM*, 2005, pp. 455–468.
[70] A. Uszok *et al.*, “Toward a flexible ontology-based policy approach for network operations using the KAoS framework,” in *Proc. MILCOM*, Nov. 2011, pp. 1108–1114.
[71] J. Bradshaw *et al.*, “The KAoS policy services framework,” in *Proc. 8th CSIIRW*, 2013, pp. 16–26.
[72] A. Voellmy, H. Kim, and N. Feamster, “Procera: A language for high-level reactive network control,” in *Proc. 1st HotSDN*, New York, NY, USA, 2012, pp. 43–48.
[73] Z. Wan, W. Taha, andP. Hudak, “Event-Driven FRP,” in *PADL: Practical Aspects of Declarative Languages*, vol. 2257, ser. LNCS. Berlin, Germany: Springer-Verlag, 2002, pp. 155–172.
[74] N. Foster *et al.*, “Languages for software-defined networks,” *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 128–134, Feb. 2013.
[75] M. Devarakonda and L. Herger, Method, apparatus, computer program product and web-enabled service providing dynamically adjustable policies, U.S. Patent 7 617 304, Nov. 10, 2009.
[76] J. Kephart, “Research challenges of autonomic computing,” in *Proc. 27th ICSE*, May 2005, pp. 15–22.

Ana Carolina Riekstin received the B.Sc. degree in computer science from the Institute of Mathematical and Computer Sciences, University of São Paulo in 2007 and the M.Sc. and Ph.D. degrees in computer engineering from the Polytechnic School, University of São Paulo in 2012 and 2015, respectively. Previously, she worked at the Laboratory of Sustainability (LASSU) / University of São Paulo, at Volkswagen do Brasil, and at Promon Logicalis. Her research interests are network management, policies, sustainability, and IT governance.

Guilherme Carvalho Januário was born in Brazil in 1987. He received the bachelor’s and master’s degrees in computer engineering from the University of São Paulo (USP), where he is pursuing the Ph.D. degree. He developed a system to translate written Portuguese to sign language (Poli-Libras) as his undergraduate project. He is interested in linguistics. In 2012, he integrated the team that developed the awarded Parliamentary Radar and joined the research group at the Laboratory of Sustainability (LASSU), University of São Paulo, where he researches optimization and sustainability in ICT, with current focus on network and device reliability and its expression based on policies.

Bruno Bastos Rodrigues received the B.Sc. degree in computer science from the University of the State of Santa Catarina (UDESC) in 2013. He is pursuing the M.Sc. degree with the Laboratory of Sustainability in TIC (LASSU), Polytechnic School, University of São Paulo. He is currently working on the “Sustainability Oriented System based on Dynamic Policies with Automated Policy Refinement” project.

Viviane Tavares Nascimento received the Bachelor’s degree in electrical engineering from the Faculty of Engineering of Ilha Solteira, Universidade Estadual Paulista (UNESP), Brazil, in 2008 and the MBA degree from Laboratory of Network Architecture (LARC), Polytechnic School, University of São Paulo, Brazil. She is currently pursuing the M.Sc. degree with the Laboratory on Sustainability (LASSU), Polytechnic School, University of São Paulo. She is currently working on the “Sustainability Oriented System based on Dynamic Policies with Automated Policy Refinement” project. She worked from July 2008 to January 2013 at Itau-Unibanco Bank.

Tereza Cristina Melo de Brito Carvalho received the B.Sc. degree in electrical engineering and the M.Sc. and the Ph.D. degrees in electronic engineering from Polytechnic School, University of São Paulo, Brazil, in 1980, 1988, and 1996, respectively. She concluded her Sloan Fellows Program (2002), as post-doctoral work, at the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. She also received a free-docency degree in electronic engineering from University of São Paulo in 2012. She specialized in the area of local network projects in the Alfried Krupp von Bohlen und Halbach Foundation, Nrnberg-Germany, in 1990. She is an Assistant Professor with the Computer Engineering Department, University of São Paulo. She serves as Director of the Laboratory of Sustainability (LASSU), University of São Paulo and is responsible for the research and development of systems in information network communication. She has more than 90 scientific and technology papers in peer reviewed journals and international conference proceedings. Her main research areas are computer networks, network management, security and on-line business.

Catalin Meirosu received the B.Sc. degree from Transilvania University, Brasov, Romania, in 1999 and the M.Sc. and Ph.D. degrees from Politehnica University, Bucharest, Romania, in 2000 and 2005, respectively. He performed his Ph.D. research at CERN, Geneva, Switzerland, while working on the data acquisition system for the ATLAS experiment at the Large Hadron Collider. He currently holds a Master Researcher position at Ericsson Research, Stockholm, Sweden, where he is focusing on management for self-organizing transport networks. Prior to joining Ericsson, he was a Project Development Officer with TERENA, Amsterdam, and a Project Associate at CERN, Geneva, Switzerland. He has co-authored more than 20 scientific articles to date. He has several patent applications pending related to his work at Ericsson. He worked on several EU-funded research projects while at CERN, TERENA, and Ericsson Research.