Rational policy system for network management

Christoph Frenzel, Henning Sanneck, Bernhard Bauer

Angaben zur Veröffentlichung / Publication details:

Frenzel, Christoph, Henning Sanneck, and Bernhard Bauer. 2013. "Rational policy system for network management." In Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), 27-31 May 2013, Ghent, Belgium, edited by Filip De Turck, Yixin Diao, Choong Seon Hong, Deep Medhi, and Ramin Sadre, 776–79. Los Alamitos, CA: IEEE. https://ieeexplore.ieee.org/document/6573076/.

Rational Policy System for Network Management

Christoph Frenzel*†, Henning Sanneck† and Bernhard Bauer*
*Software Methodologies for Distributed Systems, University of Augsburg, Germany
Email: {frenzel, bauer}@informatik.uni-augsburg.de
†Nokia Siemens Networks Research, Munich, Germany
Email: henning.sanneck@nsn.com

***Abstract*—Policy-based Network Management with Event-Condition-Action (ECA) rules is a widespread approach to face the increasing complexity in Network Management that requires a high-level of automation. However, the inability to handle uncertainty and the lack of autonomous decisions making functionality along operational objectives are limiting the applicability of ECA rules. In this paper, we present a new Rational Policy System which combines an ECA policy system with a Utility Function-based conflict resolution. In this way, it can handle the uncertainty in the environment and, at the same time, performs autonomous decision making guided by operational objectives.**

I. Motivation

Network Management (NM) is the continuous configuration of Network Elements (NEs) so that the overall network behavior and performance meets the objectives of the operator. Probes monitor and analyze the network performance data and notify the Network Management System (NMS) about detected problems via events. Triggered by the events, the NMS performs a decision making process in order to select and execute an action to handle the incident. In Policy-based Network Management (PBNM), this reasoning is guided by an operator policy. In the future, the complexity of decision making as well as the level of automation will increase, e.g., driven by heterogeneous deployments, high dynamics, and large scales. This leads to a number of challenges that NMSs will have to face [1]: among others, the system is supposed to make more complex decisions automatically. Specifically, the system should act autonomously towards achieving high-level operational objectives, e.g., maximizing network capacity with minimal operational expenses. Furthermore, the NMS needs to be able to handle uncertainty because it is not always possible to determine the current network status with certainty. This can be due to the environment, e.g., radio propagation in mobile networks, or the estimation of the network status because real-time data analysis in huge networks is impossible or impracticable. Besides that, the effects of an action are often also uncertain, e.g., an action can be ineffective in some operational context.

Traditionally, PBNM represents the policy as a set of rules which trigger actions, e.g., Event-Condition-Action (ECA) policy rules [2]. These action rules define which action satisfies the objectives of the operator the most in a specific operational situation. Hence, the reasoning how the operator's objectives can be achieved is performed by the policy developers at design-time based on their technical knowledge of the system [3]. The interweaving of objectives and technical knowledge makes the maintenance of the rules costly, e.g., if the objectives change, the policy designers have to check all rules for necessary adaptations. Furthermore, ECA-based policy systems are not well suited to handle uncertainty since they are designed for deterministic reasoning.

Researchers have developed Utility Function (UF)-based policy systems [3] which enable a higher level of automation by performing the reasoning, which actions achieve the operational objectives, autonomously. They are based on two models that make up the policy: a detailed semantic specification of the actions, e.g., logical preconditions and effects, and an explicit model of the objectives, e.g., functions which map an action and an operational network state to an abstract preference measure referred to as Utility[1]. Unfortunately, the definition of the policy, especially modeling the action effects in a complex environment, is a huge burden.

This paper presents the Rational Policy System (RPS), a new policy system that combines ECA rules with UFs. Thereby, it allows a simple policy definition and facilitates autonomous reasoning for actions given some operational objectives. Abstractly, the RPS is an ECA policy system with a UF-based conflict resolution. In that way, it can face the challenges of modern NM, i.e., the need for handling uncertainty and performing objective-based reasoning. The approach presented in this paper is a generalization of the approach in [4] and extends it with stochastic actions.

II. Concept

The concept of RPS is to extend an ECA-based policy system with a UF-based run-time conflict resolution that selects the most rational action to be performed. That means that the decision for an action is based on a policy comprising a technical system model encoded as ECA rules and operator's objectives encoded as UFs. Traditionally, an ECA policy system is supposed to strictly obey the policy rules in order to satisfy the operator's objectives. Hence, the rules mix up technical knowledge and operational objectives. In the RPS, ECA policy rules solely encode which actions might be an appropriate response to an event in some operational context from a technical point of view, i.e., they encode the technical knowledge about the network. The operational objectives are then considered during the selection of one of the applicable actions, i.e., conflict resolution.

The RPS reacts to events that are raised by monitoring and analysis probes in order to notify the NMS about some detected and diagnosed issue, e.g., a defect of an NE or a load imbalance. The events are annotated with a Probability that refers to the likelihood that the issue that is indicated by the

[1]Policies using UFs with binary codomain are often referred to as goal policies. A detailed discussion of different policy types can be found in [3].

event is actually present, e.g., the probability that an NE is broken. The probes may utilize sophisticated diagnostic methods like event correlation or Bayesian reasoning which allow the estimation of root cause probabilities from symptoms. Notice that it is assumed that the RPS can decide for an action to handle an event without any additional root cause analysis.

The applicable actions, proposed by the ECA rules for a set of events, are very likely to be in conflict in the RPS. This is due to a number of reasons: first, classical policy conflict reasons like modality, e.g., turn-on and turn-off of the same NE, or application-specific constraints, e.g., energy-saving mode for an NE which should also take over the load of another NE [5]. Second, the ECA policy rules solely encode technical knowledge, i.e., no design-time conflict resolution based on the operational objectives has been performed. Hence, it is likely that several rules will fire for an event and propose actions. Third, since the probes can represent the uncertainty in their analysis, they do not have to consider fixed certainty threshold that are usually set to a high value to avoid false events. Instead, the probes will raise numerous events and the RPS will decide dynamically which of these are handled.

The conflict resolution in the RPS is based on decision theory. It resolves a conflict by executing the most rational action, i.e., the action that fulfills the operational objectives the most in the uncertain situation. The operational objectives are "persistent, abstract, user-oriented objectives for how a system should behave" [6], i.e., they are describing what the system should achieve in a declarative way. Decision theory, which has been studied, e.g., in Artificial Intelligence (AI), provides a consistent framework for making a rational decision for one of several options [7]. In essence, it defines a measure for the degree of rationality of an action, here referred to as Value.

In NM, the Value of an action depends on four factors:

1) The action Cost refers to a measure of the operator's preference for an action.
2) The action Effectiveness refers to the probability that the action will produce the indented effects, i.e., it handles an event by treating the indicated problem. This is necessary since we consider stochastic actions.
3) The event Utility refers to a measure of the operator's preference that an event is handled.
4) The event Probability.

Although the Cost and Utility are simple measures, i.e., usually real numbers, they can also represent complex preference systems with multiple dimensions. So besides a monetary value, the Cost can also express qualitative aspects like the action's impact on network performance. In parallel, the Utility can represent the severity of the incident indicated by an event, e.g., the decrease of network performance from the optimum. This can be achieved by defining the Cost and Utility to be a weighted sum of the individual degrees to which the objectives in each dimension have been fulfilled.

III. System Design

This section outlines the design of an implementation of the presented RPS concept. The system is assumed to react at regular points in time to an event set, i.e., a set of events and their associated Probabilities that have been

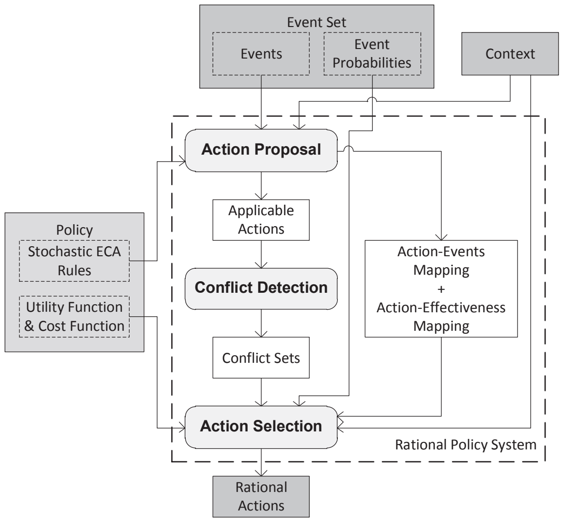

Figure 1. The Rational Policy System with Action Proposal, Conflict Detection, and Action Selection.

raised by the monitoring and analysis probes between two runs. Additionally, the system can access operational context information which contains configuration data about the network, e.g., the current date and time, the network topology or the network configuration. In essence, the context provides all information that is necessary to evaluate the conditions of the technical ECA rules. Notice that the RPS assumes certainty in the context information, since network configuration data is usually known with certitude.

No specific spatial scope of the event set or the context is assumed. For instance, if the RPS manages a network centrally then the event set would contain events from the whole network and the context would provide information about the whole network. In contrast in a distributed system, the events and the context would be from a single NE.

The reasoning of the RPS at run-time is performed in three steps as shown in Figure 1:

1) The Action Proposal (AP) evaluates the stochastic ECA rules in order to determine applicable actions for the events in the event set.
2) The Conflict Detection (CD) analyzes the set of applicable actions for conflicts.
3) The Action Selection (AS) resolves each conflict by selecting the most rational action, i.e., the one with the highest Value, to be executed.

A. Action Proposal

The AP determines the set of applicable actions for the events in the event set in the given context. Thereby, it does not consider the uncertainty in the event set, i.e., it neglects the event Probabilities. The reasoning is based on a set of stochastic ECA policy rules which have the form:

ON event IF condition THEN action WITH effectiveness

In this policy, event refers to an event which triggers the policy rule, condition is a logical expression over the context, and action is the action which should be triggered if the event is raised and the condition is true. Besides these classic components of ECA rules, stochastic ECA rules also have an effectiveness: it refers to the probability that the action resolves the problem indicated by event under the condition, i.e., $P(\text{action is effective}|\text{event}, \text{condition})$. For instance, the following ECA policy states that if the software of an NE crashed and the software is outdated then a software update resolves the problem with probability of 0.7:

ON software crash
IF installed software version < current software version
THEN update software WITH 0.7

Besides determining the applicable actions, AP creates two mappings: the action-event mapping encodes the events treated by an applicable action and the action-effectiveness mapping encodes the Effectiveness of an applicable action for an event.

B. Conflict Detection

The CD is responsible for analyzing the applicable actions for potential conflicts (cf. Section II). Policy conflict detection is a complex task and an active research community developed numerous approaches [5], [8], [9]. The RPS is not dependent on a specific policy conflict detection approach. However, the CD is required to produce a set of conflict sets, i.e., sets containing actions that are in conflict with each other, from the set of applicable actions.

C. Action Selection

For each conflict set, the AS decides which action is the most rational one to be executed given the operational context. As depicted in Figure 1, the AS relies on the operator's objectives that are encoded as a Cost and a Utility function.

The Cost function defines a Cost for all possible actions given a specific context. Thereby, the lower the Cost of an action in a context, the more preferred it is. In this way, it is possible to define, e.g., that a maintenance restart of an NE is more preferred at night than during the day. Consequently, the Utility function determines a Utility for all events in a specific operational context. However, in contrast to the Costs, the handling of an event is more preferred if its Utility is high.

The AS is a process that iterates over all conflict sets. In the first step, the Values of all actions in a conflict set are computed. Thereby, a greedy approach is taken, i.e., only the immediate Utilities that an action produces are considered. Although it is not necessarily globally optimal, it provides good results with a reasonable computational complexity. The Value V of an action is the difference of the expected Utility of the action U_{exp} and the action's Cost C:

$$V = U_{\text{exp}} - C \tag{1}$$

The expected Utility of an action is the sum of the expected Utilities of each possible action outcome, i.e., the possibly present problems that the event set indicates and that the action can handle. Furthermore, the expected Utility of a possible action outcome is the sum of the Utilities of the treated events weighted with the probability of that outcome. Since we assume that the issues of multiple events can be actually present, it would be necessary to consider each possible combination of treated events as an outcome. However, by assuming that the combined Utility of an outcome is the sum of the Utilities of the actually treated issues, the calculation of the expected Utility can be simplified: the expected Utility of an action is the sum of the expected Utilities for each single event that the action treats. That is, the expected Utility U_{exp} of an action is the sum of the Utilities U of the treated events E weighted with the event Probability P and the action Effectiveness P_{eff}:

$$U_{\text{exp}} = \sum_{e \in E} P_e \cdot P_{\text{eff}} \cdot U_e \tag{2}$$

In the second step, the AS selects the most rational action from the conflict set to be executed. According to decision theory, the most rational action is the one with the highest Value. Thereby, if no action would be performed then the Value of of this decision is 0 since this imaginary action has no Cost and produces no Utility because it treats no event. Hence, the AS picks the action with the highest positive Value or no action if all actions have a negative Value. After the execution of these steps for all conflict sets, the AS returns the set of rational actions which should be executed in the network.

IV. Policy Creation

In order to use the RPS, the two models of the policy have to be created. In order to design the stochastic ECA policy rules, the operators can use ordinary ECA policy rules that might be already present in the NMS as a starting point. First, these rules need to be revised. Especially the rules and conditions that are the result of a reasoning about the operational objectives need to be removed. Second, the Effectiveness needs to be determined. This can be quite costly since operators are not familiar with the specification of probabilities. However, it is possible to start with an initial estimation, e.g., all rules have an Effectiveness of 1, and later refine it.

The elicitation of the operator's preferences in form of Utility and Cost functions is not trivial [3]. Usually, they are, if at all, vaguely communicated. Therefore, a practical approach is to set the Utilities of the events initially to an equally high value whereas the Costs for the actions are equally low. In this way, the system reacts to an event set with the action that is most likely to handle the events without any preferences. Later, when the operator develops some experience with the system's operation, the Costs and Utilities can be adapted in order to be closer to their true values.

V. Case Study

In order to illustrate the RPS concept, this section presents an application scenario that is inspired by [4]. It describes a simplified failure recovery system for a 3rd Generation Partnership Project (3GPP) Long Term Evolution (LTE) mobile network. Figure 2 depicts the policy rules, Utilities, and Costs of this scenario. The events are indicating problems, which have been diagnosed by probes in the network and the policy set describes which actions possibly mitigate or resolve a problem. Furthermore, the Utilities refer to the severity of the failures and the Costs represent the costs and impacts of the actions on normal network operation. For instance, the policy ON txp problem IF context(time=night) THEN restart ne

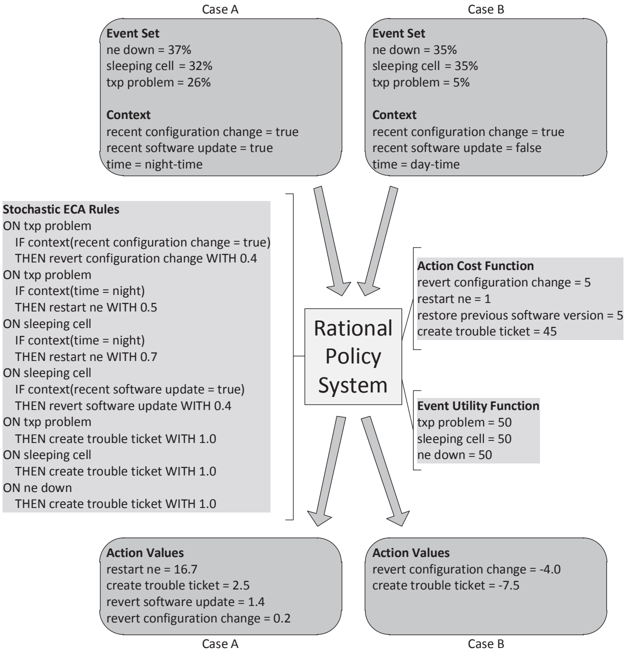

Figure 2. The policy, the event sets, and Values for Case A and Case B of the application scenario (following [4]).

WITH 0.5 states that the restart of an NE might help with an Effectiveness of 0.5 if a txp problem, i.e., a problem with the transmission power, has been diagnosed at night.

Especially in complex scenarios, the RPS can produce subtle behavior, as can be seen in the two example cases, Case A and Case B, in Figure 2. Case A shows a high probability that a broken NE is present. Although this problem can solely be recovered with a trouble ticket, the RPS decides to try a restart of the NE first since the trouble ticket is much less preferred than a restart. Case B depicts another situation which is characterized by low probabilities for all root causes. In this case, the RPS suggests to perform no action at all since all actions have negative Values. However, at a later point in time, e.g., at night, this might change.

VI. Related Work

Introducing uncertainty into rule-based systems is not a new idea: for instance, FuzzyCLIPS [10] allows to assign certainty factors to both facts and rules. However, these approaches have no notion of Value.

Baliosian et al. [9] and Bahati and Bauer [11] presented policy systems which resolve policy conflicts at run-time based on the effectiveness of the applicable actions. This effectiveness is related to the Effectiveness of the stochastic actions in this paper, and even extends it with machine learning. However, their shortcoming is that they neither consider uncertainty in the events nor complex operational objectives.

There are also approaches which introduce the consideration of operational objectives into policy systems. For instance, Aib and Boutaba [12] presented a system which relies on a complex, mathematical queuing model in order to simulate the effects of an action and calculate its Value. Unfortunately, such a model is hard to develop for real networks. ACCENT [6], however, is based on an model that explicitly expresses the effects of the actions. This approach seems to be easier to implement since the effects are known to the operators. Nevertheless, both systems do not consider uncertainty in either the events or the results of the actions.

VII. Conclusion

Traditional NM approaches are challenged by the increasing pressure for automation. The presented RPS combines an ECA policy system with a UF-based conflict resolution in order to face these challenges. In this way, it can handle the uncertainty in the environment and, at the same time, performs autonomous decision making guided by operational objectives.

The elicitation of the UFs is a huge burden and, so, more research is required on the modeling and verification of the Costs and Utilities especially for complex, multidimensional preference systems. Furthermore, it is promising to extend the RPS with machine learning to estimates the Effectiveness of the stochastic actions autonomously and extend the context to include uncertain information as well.

References

[1] A. Pras, J. Schonwalder, M. Burgess, O. Festor, G. Perez, R. Stadler, and B. Stiller, "Key research challenges in network management," *IEEE Communications Magazine*, vol. 45, no. 10, pp. 104–110, Oct. 2007.

[2] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation.* Amsterdam; Boston: Morgan Kaufmann, 2004.

[3] J. Kephart and W. Walsh, "An artificial intelligence perspective on autonomic computing policies," in *Proc. IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2004)*, Yorktown Heights, USA, Jun. 2004, pp. 3–12.

[4] C. Frenzel, H. Sanneck, and B. Bauer, "Automated rational recovery selection for self-healing in mobile networks," in *Proc. International Symposium on Wireless Communication Systems (ISWCS 2012)*, Paris, France, Aug. 2012, pp. 41–45.

[5] E. Lupu and M. Sloman, "Conflicts in policy-based distributed systems management," *IEEE Transactions on Software Engineering*, vol. 25, no. 6, pp. 852–869, 1999.

[6] K. J. Turner and G. A. Campbell, "Goals and Conflicts in Telephony," in *Proc. International Conference on Feature Interactions in Software and Communication Systems (ICFI 2009)*, no. June, Lisbon, Portugal, Jun. 2009, pp. 3–18.

[7] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[8] R. Boutaba and I. Aib, "Policy-based Management: A Historical Perspective," *Journal of Network and Systems Management*, vol. 15, no. 4, pp. 447–480, Nov. 2007.

[9] J. Baliosian, K. Matusikova, K. Quinn, and R. Stadler, "Policy-based self-healing for radio access networks," in *Proc. IEEE Network Operations and Management Symposium (NOMS 2008)*, Salvador, Bahia, Brazil, Apr. 2008, pp. 1007–1010.

[10] R. A. Orchard, "FuzzyCLIPS Version 6.04A - User's Guide," Institute for Information Technology, National Research Council Canada, Tech. Rep., Oct. 1998. [Online]. Available: http://awesom.eu/~cygal/FuzzyCLIPS/fzdocs.pdf

[11] R. M. Bahati and M. A. Bauer, "Reinforcement learning in policy-driven autonomic management," in *Proc. IEEE Network Operations and Management Symposium (NOMS 2008)*, Salvador, Bahia, Brazil, Apr. 2008, pp. 899–902.

[12] I. Aib and R. Boutaba, "Business-Driven Optimization of Policy-Based Management solutions," in *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM 2007)*, Munich, Germany, May 2007, pp. 254–263.