Automated Rational Recovery Selection for Self-Healing in Mobile Networks

Christoph Frenzel
Programming Distributed Systems
University of Augsburg, Germany and
Nokia Siemens Networks Research
Munich, Germany
Email: frenzel@informatik.uni-augsburg.de

Henning Sanneck
Nokia Siemens Networks Research
Munich, Germany
Email: henning.sanneck@nsn.com

Bernhard Bauer
Programming Distributed Systems
University of Augsburg, Germany
Email: bauer@informatik.uni-augsburg.de

***Abstract*—Self-healing is a key functionality of Self-Organizing Networks (SON). There have already been promising research results on Degradation Detection and Root Cause Diagnosis. However, the complex task of Recovery Selection, i.e., the determination of the best recovery action for an uncertain diagnosis of a degradation considering the operational goals of the network operator, is rarely investigated so far. In this paper, a two-step rational Recovery Selection system and its integration into a self-healing process for mobile networks is presented. Thereby, Recovery Selection first exploits technical recovery knowledge expressed in rules to determine the set of possible recovery actions for a degradation situation. Second, it determines the degree of rationality of these actions based on the operational goals of the operator. Therefore, it draws on decision theory in order to handle the uncertainty in the problem situation and the possibly conflicting preferences of the operator. The presented system enables automatic rational Recovery Selection which provides a baseline for self-healing in mobile networks.**

I. Introduction

The 3rd Generation Partnership Project (3GPP) SON is an approach to automate mobile network operations through self-configuration, self-optimization, and self-healing [1]. Self-healing, also referred to as automated troubleshooting, promises to significantly reduce human workload in troubleshooting, which is a major cost factor especially in the Radio Access Network (RAN). However, it is a challenging research area due to the uncertainty in the physical environment and the complexity of modern cellular networks.

The 3GPP defined a SON self-healing reference process which distinguishes between the detection of a problem, the determination of the problem's root causes and recovery actions, and the execution of the actions [2]. However, this work does not recognize the need for an explicit analysis to determine the best recovery action for the diagnosed root causes considering the operational goals of the network operator. Adding such a capability leads to the four-step self-healing process depicted in Figure 1, consisting of *Degradation Detection*, *Root Cause Diagnosis*, *Recovery Selection (RS)*, and *Action Execution*.

The Degradation Detection step identifies degraded network resources based on a detection model and triggers the Root Cause Diagnosis, which determines the probabilities of the existence of different root causes based on a diagnosis model.

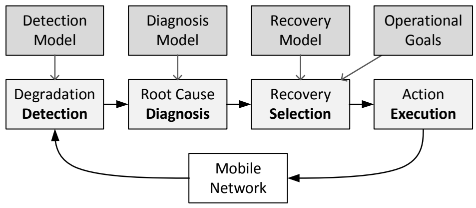

Fig. 1. Four-step self-healing process for mobile networks.

Given the result of the diagnosis, RS determines possible actions to recover from the degradation based on a recovery model and makes a rational decision which action to execute given the operational goals. Finally, the execution of the selected action is performed by the Action Execution. In case the action does not make the degradation disappear, the self-healing process is reiterated. Although this process is supposed to run automatically, it can also be used in an expert system mode, i.e., it presents the most rational recovery action to a human operator who takes this information into account during manual troubleshooting.

There are several interesting and promising research results on the automation of the detection and diagnosis steps [3]–[5]. The developed mechanisms can provide a human operator with valuable information and, so, ease troubleshooting. Nevertheless, in order to enable self-healing, RS has to be automated as well. However, there has been little research on this topic in the past. One reason for this lack of interest might be that RS is often seen as a simple mapping between a root cause and a recovery action. However, state-of-the-art probabilistic diagnosis approaches, e.g., [3], provide multiple probable root causes which can be recovered in several ways. In order to make rational decisions in this setting, RS has to be more sophisticated in order to handle the uncertainty in the diagnosis result, lots of contextual information about the network, and the operational goals. For instance, restarting an Network Element (NE), e.g., a radio base station, usually resolves software induced problems, however, at the cost of a temporary outage. Therefore, it might be preferred to try a configuration change before although this is less likely to recover the system.

This paper presents an automated recovery selection system for self-healing in the RAN based on rules and decision theory. On the one hand, rules are used to encode technical knowledge about the dependencies between actions and root causes. On the other hand, decision theory is used to determine the degree of satisfaction of the operational goals by a specific recovery action given an uncertain diagnosis result. Although the approach is not dependent on a particular technology, the focus of this paper lies on 3GPP Long Term Evolution (LTE) networks. The system enables automatic reasoning for the most rational recovery action in a specific problem situation and, thus, facilitates self-healing in mobile networks.

II. Rationality in Recovery Selection

Decision theory is a research topic concerned with the analysis of the effects of decisions in order to evaluate the usefulness of different options for a decision maker [6]. It draws on probability theory for handling uncertainty and on utility theory for allowing the expression of preferences. Based on this, the degree to which a particular recovery action is rational in a specific degradation situation can be determined by a measure called Utility.

For the computation of the action Utility in mobile networks troubleshooting, three dimensions need to be considered:

1) the probability that the action will be effective,
2) the preferences regarding the recovery action, and
3) the preferences regarding the degradation resolution.

The probability that an action is effective is directly related to the probability of the root causes it resolves. Note that a specific recovery action is often applicable to several problems. For instance, issuing a trouble ticket might make all problems vanish since an expert troubleshooter takes care of the problem. Restarting an NE, on the other hand, may solely treat the phenomenon of sleeping cells, i.e., non-operational network cells which do not raise any alarm [5].

The preferences of an operator regarding recovery actions can be diverse and numerous. For instance, operators may prefer fast actions to resolve a degradation quickly, automatic actions to minimize the cost of troubleshooting, or actions which are executed at low-traffic hours to reduce the impact of troubleshooting on the mobile network. However, the preferences are often conflicting with each other. For instance, a configuration change of an NE takes longer than a restart since a lot of additional measurements have to be taken in order to determine the new settings. However, a configuration change usually has a smaller impact on network operation.

The operator's preferences for an action also depend on whether this action has already been executed and did not resolve the current degradation situation. On the one hand, the repetition of ineffective actions should be avoided but, on the other hand, it can be desired to try an action several times because it is not always effective. For instance, a NE restart is often performed several times before a trouble ticket is issued.

Besides executing recovery actions, the system can also trigger an observation, i.e., the collection further measurements in order to make the diagnosis result more conclusive. The rationality of the observation action does not depend on the resolved root causes but instead on the benefit of the additional information the system can gain through it.

The preferences regarding the resolution of a particular degradation can be as complex as the action preferences, e.g., minimizing the reduction of capacity or minimizing the reduction of coverage. By taking them into consideration, the RS is able to decide to postpone a recovery action and let the problem stay untreated. For instance, a reduced coverage area of a network cell might have little impact in a high density network with low traffic load. However, triggering a restart of the cell at day-time has a high impact because active calls in the cell are dropped. Hence, it is more rational to postpone the restart until its impact is less severe, e.g., at night.

Rational RS chooses an action which maximizes the Utility over the three dimensions. This can lead to subtle behavior.

III. Related Work

Wille et al. [3] were among the first to acknowledge the need for a rational RS and provide a discussion about the selection of recovery actions based on root cause probabilities and action costs, i.e., their negative Utilities. They propose the application of a decision-theoretic troubleshooting approach like the generic rational troubleshooting process by Heckerman et al. [7]. This framework determines a sequence of recovery actions and observations for an uncertain diagnosis result, which minimizes the overall expected cost. Although it provides a good basis, the application of the approach to RS in mobile networks is limited: the cost measure is restricted to actions and does not consider preferences regarding the degradation resolution; the applicability of actions cannot be constrained, e.g., an NE restart must not be performed during office hours; and a recovery action resolves only one problem.

Baliosian et al. [8] propose a rule-based system, Omega, for RS. The rules, which encode the recovery model, can be in conflict in a specific failure situation, i.e., several actions are proposed. The selection of a single action is based on a simple cost measure as well as a learned probabilistic model of the action's effectiveness. However, the framework does neither consider the uncertainty in the diagnosis result nor preferences regarding the resolved problems.

In the context of the Self–NET project, a framework for decision making in Future Internet network elements has been developed [9]. It is based on a fuzzy rule system and enables modeling of recovery rules for an uncertain environment. Furthermore, it allows hierarchical decision making. The shortcoming of the framework is that it solely handles uncertainty but does not consider any operational goals. Thus, it cannot be seen rational.

IV. Rational Recovery Selection System

RS is performed in a two step process as shown in the center of Figure 2. First, Recovery Planning (RP) determines the applicable actions for the diagnosis result given the recovery model. As a result, it produces a resolution mapping between actions and the root causes they resolve. The subsequent

Recovery Decision Making (RDM) takes this mapping and the diagnosis result, and determines the Utility of each action based on the operational goals.

Both process steps consider the operational context during reasoning. The context is a collection of data from configuration management, e.g., the type of NE, data from performance management, e.g., the current number of users, and other information, e.g., date and time. This allows the restriction of the execution of actions and the adaption of the operator goals to the network resource and its environment.

A. Recovery Planning

RP exploits technical troubleshooting knowledge of operators, expressed in the recovery model, to find applicable actions for a given set of diagnosed root causes without considering their probabilities. Formally, RP determines the set of recovery actions A[1] and observations O which are applicable for the given set of root causes Φ and a context $\chi \in X$. Furthermore, it produces a resolution mapping $R : A \cup O \mapsto \mathcal{P}(\Phi)$ expressing the effects of a recovery action $\alpha \in A$ or observation $o \in O$, respectively, by mapping it to the set of treated root causes.

Various formalisms can be used to represent the recovery model, e.g., logic-based, case-based, or decision tree-based approaches. However, a rule system seems to be an advantageous option for two reasons: first, rules are widely used and well understood in network management [10] and, second, there are already rule-based systems for troubleshooting in operation which can be adapted to be used in RP. This allows to acquire the information for building the recovery model from automated systems which are already in place, troubleshooting handbooks, or expert knowledge in a simplified way.

The recovery model is defined using obligation rules of the form IF condition THEN action. Thereby, condition is a logical combination of root causes and operational context properties. It can be omitted, e.g., to define a generic recovery action trouble ticket applicable to all problems. For instance, IF sleeping cell AND cell area = rural area THEN reset NE proposes an NE reset if a sleeping cell in a rural area is diagnosed. Depending on the action type, i.e., a recovery action or an observation, the rule semantics vary. Consider the rule IF r THEN a: if a is a recovery action then a potentially resolves r, whereas if a is an observation action then a can potentially provide further confidence in the presence of r. Additionally, prohibitive rules allow to forbid the execution of an action: IF condition THEN NOT action. For instance, the following rule does not allow the execution of an action if an engineer is at the site: IF engineer at site = true THEN NOT any action.

The recovery model describes a mapping from root causes, expressed in the conditions, to actions. Thus, RP can determine the applicable actions and observations by executing the model using a rule engine. In order to create the reversed resolution mapping R, however, it is necessary to monitor the execution of the rules and track the dependencies between each action and the respective root causes.

[1]Introduced symbols will be used throughout the paper.

B. Recovery Decision Making

RDM determines the Utility of each applicable action based on the operational goals and selects the one with the highest Utility, i.e., the most rational one.

The operational goals regarding the actions are expressed by using the abstract measure *Cost* which represents the degree to which a recovery action or observation deviates from the operator's preferences. Hence, the lower the Cost, the more preferred the action. Formally, it defines a function $C : (A \cup O) \times X \mapsto \mathbb{R}$ which maps an action or observation, respectively, and a context to a real number, i.e., the Cost. In order to make a trade-off between multiple preferences, the Cost is usually calculated as the weighed sum over the degrees to which the action satisfies the different preferences [6]. Furthermore, the Cost must increase with the repeated execution of an action in order to make it less preferable. This can be easily incorporated by adding a penalty to the Cost of an action in proportion to the number of repetitions, which can be extracted from the context X. However, an observation will never be repeated.

In a similar way, a function $V : \Phi \times X \mapsto \mathbb{R}$ is defined for the root causes. V defines the *Value* of a root cause $\varphi \in \Phi$ which represents the degree to which the resolution of it is preferred by an operator. Consequently, it holds that the higher the root cause's Value, the more preferred its resolution.

Based on the operational goals, the diagnosis result, and the resolution mapping from RP, RDM can determine the Utility of the applicable actions. The Utility of a recovery action in a specific operational context is the sum of the negative Cost of the action and the expected Value it produces, i.e., the Value of all root causes that the action resolves:

$$U(\alpha, P, \chi) = -C(\alpha, \chi) + \sum_{\varphi \in R(\alpha)} P(\varphi) V(\varphi, \chi) \quad (1)$$

where $P(\varphi)$ is the normalized probability of root cause φ in the diagnosis result, i.e., $\sum_{\varphi \in \Phi} P(\varphi) = 1$. It is easy to see that actions with lower Cost, or actions that resolve a severer and more probable degradation have a higher Utility. Notice that the Utility of an assumed action do nothing, which resolves no root cause at all, would be 0. Hence, all actions with a Utility higher than 0 are potentially rational.

The Utility of an observation action o is calculated differently since it does not resolve any root causes and, hence, does not directly produce a Value. Instead, they are gathering information that the system can use to potentially make better decisions. This is referred to as the value of information [6]:

$$U(o, P, \chi) = -C(o, \chi) + U^{+}(o, P, \chi) + U^{-}(o, P, \chi) \quad (2)$$

computes the Utility of an observation as the sum of the negative Cost of the observation and the Utility of the positive observation $U^{+}(o, P, \chi)$ as well as negative observation $U^{-}(o, P, \chi)$. The positive observation Utility is the expected

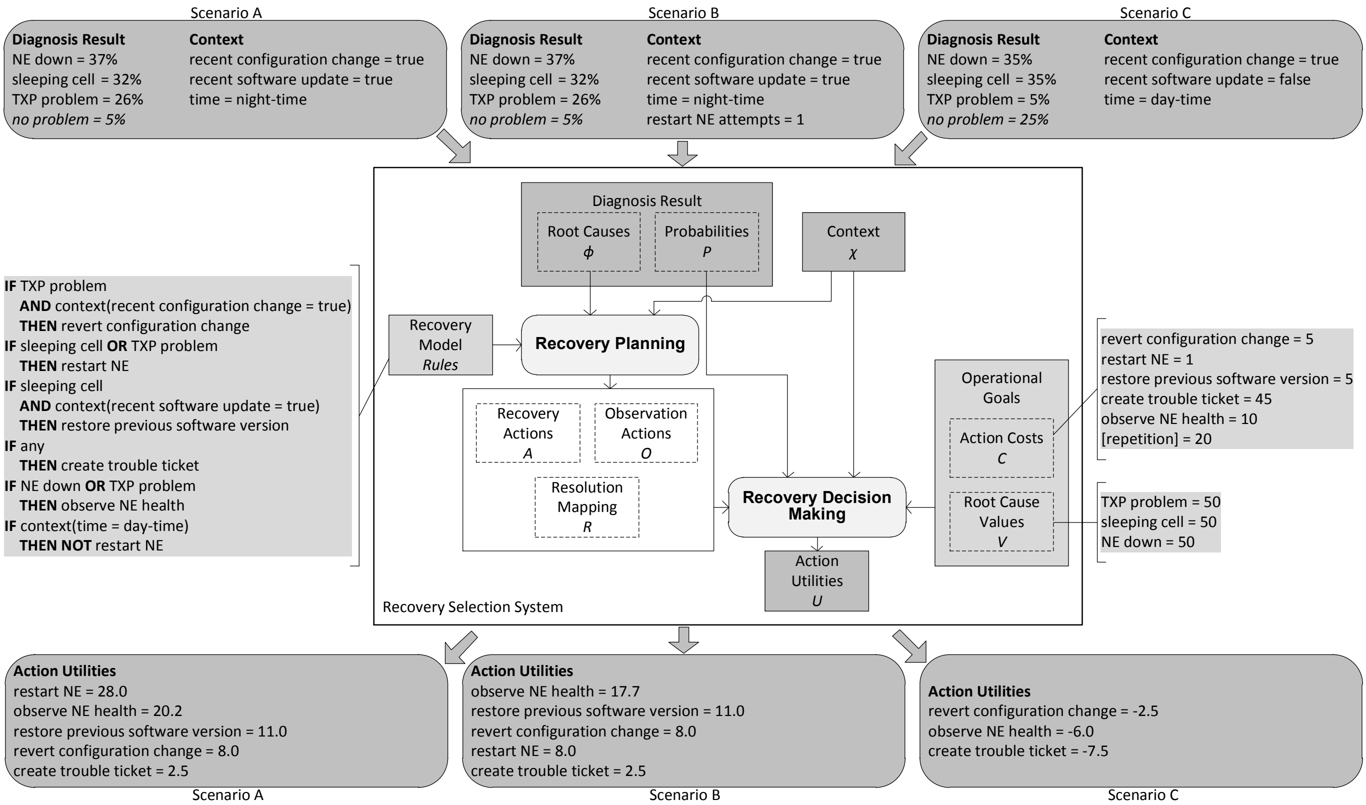

Fig. 2. The rational RS system together with the recovery model, operational goals, and scenarios from the case study.

maximal Utility of all recovery actions if the observation confirms one of its mapped root causes:

$$U^{+}(o, P, \chi) = \sum_{\varphi \in R(o)} P(\varphi) \max_{\alpha \in A} U\left(\alpha, P^{+}_{\varphi}, \chi\right) \tag{3}$$

where P^{+}_{φ} refers to a normalized probability function with

$$P^{+}_{\varphi}(r) = \begin{cases} 1 & \text{if } r = \varphi \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The negative observation Utility is the expected maximal Utility if all assigned root causes are refuted by the observation:

$$U^{-}(o, P, \chi) = \left(1 - \sum_{\varphi \in R(o)} P(\varphi)\right) \max_{\alpha \in A} U \quad \alpha, P^{-}_{R(o)}, \chi \tag{5}$$

where $P^{-}_{R(o)}$ refers to the normalized probability function, i.e., $\sum_{\varphi \in \Phi} P^{-}_{R(o)}(\varphi) = 1$, proportional to the unnormalized measure

$$\widetilde{P}^{-}_{R(o)}(r) = \begin{cases} P(r) & \text{if } r \notin R(o) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The presented decision making approach makes two assumptions on the troubleshooting process. First, only one of the diagnosed root causes is actually present in the network. This *single fault assumption* eases troubleshooting and is used in several Root Cause Diagnosis approaches, e.g., [3]. If it does not hold then the Utilities may be incorrect and, hence, RS not optimal. Second, the diagnosis result has to contain the probability of a no problem case, i.e., a misdetected degradation. This enables the system to skip a costly recovery attempt if the problem is unlikely.

Equation 1 encodes an One-Shot Decision Process (ODP), i.e., it solely considers the direct Costs and Values of an action [6]. In contrast, the Sequential Decision Process (SDP) in [7] also considers the Utilities of subsequent decisions if an action is not successful in resolving the problem. Although SDP seems to be more accurate at first sight, in the context of mobile networks troubleshooting, it is an approximation: since the duration of an action can be minutes or hours, the system is likely to be in a different state after the execution due to, e.g., the progression of time and changes in user behavior. In the following state, the set of applicable actions, the Costs, and Values might be different. Unfortunately, this transition is barely predictable. Given this fact and the increased complexity of SDP, ODP seems to be reasonable choice.

The quality of RDM depends on the operational goals which are, unfortunately, hard to elicit since they are often not explicitly communicated. Besides complex methods like the Analytic Hierarchy Process [11], one can also mitigate this problem with a smooth transition approach: initially, all root causes are assigned high Values and actions have small Costs. So, the resolution of a degradation has a high priority without any preference for a particular action or root cause. Consequently, the system will select the action with the highest probability of effectiveness. Later, the Values and Costs can be adapted in order to align the model with the actual preferences.

V. Evaluation

The rational RS approach has been implemented in an experimental system in order to evaluate its feasibility as well as the presence of complex behavior.

A. Experimental System

RP uses JBoss Drools Expert [12] as the rule system because it provides a simple and human readable syntax for expressing both obligation and prohibitive rules for the recovery model. The Costs, Values, and the operational context are provided as key–value pairs. At run-time, the system allows the creation of artificial diagnosis results which are subsequently fed into RP and RDM. As a result, the system presents the Utilities of all applicable actions.

B. Case Study

The baseline for the case study is a semi-automatic troubleshooting process in a 3GPP LTE mobile network including detection and diagnosis, which is extended with the RS system. For this purpose, it is necessary to define the recovery model and the operational goals for the network which can be hard to elicit. Fortunately, as mentioned in Section IV-A and Section IV-B, there are several possible sources of information and methods supporting this process. Figure 2 shows the two models that are used in this study: on the left, there are the rules that make up the recovery model and, on the right, there are the operational goals comprising Costs and Values. Thereby, [repetition] refers to the penalty for a repeatedly executed action.

For the evaluation, several degradation scenarios, i.e., prepared diagnosis results and context descriptions, were fed into the system. Three of these are shown at the top in Figure 2, namely Scenario A, B, and C. After the execution, the computed Utilities of the actions, shown at the bottom in the same figure, were collected and analyzed.

C. Results

The three evaluation scenarios are sufficiently complex that subtle behavior as mentioned in Section II can be observed.

In Scenario A the most probable root cause for the problem is a broken NE which can be recovered solely by issuing a trouble ticket. This is expensive, tough. Thus, although a sleeping cell is much less likely, the system suggest to perform a restart because it is relatively cheap.

Suppose that the executed restart was not effective, i.e., the problem still exists as shown in Scenario B. Then the context contains the information that a restart has already been performed and, so, the Utility of this action is penalized. As a result, an additional observation seems most reasonable now. Notice that the Utility of the observation action decreased compared to Scenario A. This is because after the observation, a restart still has a smaller Utility than in Scenario A. Notice that in Scenario B, it is assumed that the Root Cause Diagnosis does not consider the information about the ineffective action.

Scenario C finally depicts the case that the system suggest to do nothing, i.e., postpone recovery, by assigning negative Utilities to all actions. This is because all root causes are quite unlikely and a cheap restart is not possible during the day.

VI. Conclusion

Recovery Selection in mobile networks, i.e., making the decision for one recovery action given an uncertain diagnosis of a degradation and operational goals, is a complex process. This paper presents a two-step rational Recovery Selection system: first, *Recovery Planning* determines the applicable recovery actions in a specific degradation situation based on recovery knowledge expressed as rules. Second, *Recovery Decision Making* computes the degree of rationality of the actions, i.e., their Utility, drawing on the profound decision theory. The system goes beyond related work in the sense that it is the first Recovery Selection approach that specifically targets the special requirements of mobile networks troubleshooting, e.g., complex operational goals. By integrating the system into a four-step self-healing process, this paper provides a baseline for self-healing in mobile networks.

In the future, the modeling of stochastic actions, i.e., actions which resolve a problem with some probability, seems to have the potential to increase the accuracy of Recovery Selection. Based on this, it is possible to employ machine learning techniques to estimate the required probabilities. Furthermore, the elicitation and validation of the operational goals requires further investigation: they are an abstract and indirect description of a recovery process and, so, the prediction of system behavior is hard. Nevertheless, mobile network operators need to be assured that self-healing performs as expected.

References

[1] S. Hämäläinen, H. Sanneck, and C. Sartori, Eds., *LTE self-organising networks (SON)*. Chichester: John Wiley & Sons, 2011.

[2] 3GPP, "Self-healing concepts and requirements (release 10)," Technical Specification TS 32.541 V10.0.0, Mar. 2011.

[3] V. Wille, S. Patel, R. Barco, A. Kuurne, S. Pedraza, M. Toril, and M. Partanen, "Automation and optimisation," in *GSM, GPRS and EDGE Performance*, T. Halonen, J. Romero, and J. Melero, Eds. Chichester: John Wiley & Sons, 2004, ch. 13.

[4] S. Nováczki and P. Szilagyi, "Radio channel degradation detection and diagnosis based on statistical analysis," in *Proc. VTC Spring 2011*, Budapest, Hungary, May 2011.

[5] SOCRATES, "D5.9: Final report on self-organisation and its implications in wireless access networks," Deliverable INFSO-ICT-216284 SOCRATES, Jan. 2010.

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

[7] D. Heckerman, J. S. Breese, and K. Rommelse, "Decision-theoretic troubleshooting," *Commun. ACM*, vol. 38, pp. 49–57, Mar. 1995.

[8] J. Baliosian, K. Matusikova, K. Quinn, and R. Stadler, "Policy-based self-healing for radio access networks," in *Proc. NOMS 2008*, Salvador, Brazil, Apr. 2008.

[9] Self-NET, "D2.3: Final report on self-management artefacts," Deliverable ICT-2007-224344/Self-NET/WP2/D2.3/31042010, Apr. 2010.

[10] J. Strassner, *Policy-based network management*. Amsterdam; Boston: Morgan Kaufmann, 2004.

[11] T. L. Saaty, *The analytic hierarchy process*. New York; London: McGraw-Hill, 1980.

[12] Drools expert. [Online]. Available: http://www.jboss.org/drools/drools-expert