

# Policy-driven Workflows for Mobile Network Management Automation

Raphael Romeikat,  
Bernhard Bauer  
University of Augsburg  
Augsburg, Germany  
{romeikat,bauer}  
@ds-lab.org

Tobias Bandh,  
Georg Carle  
TU München  
Garching, Germany  
{bandh,carle}@in.tum.de

Henning Sanneck,  
Lars Christoph Schmelz  
Nokia Siemens Networks  
Munich, Germany  
{henning.sanneck,  
lars.schmelz}@nsn.com

## ABSTRACT

Future wireless networks will experience a continuous growth regarding the number of network elements with increasingly complex interrelations between the configuration of multiple network elements. Another trend is the seamless integration of multiple radio technologies into a single wireless network. Both developments increase network management complexity and require new management concepts with a very high degree of automation. For this purpose, a novel management approach based on a combination of workflow and policy technologies is presented. The goal is to simplify and automate management tasks in mobile networks in order to raise the state of self-organization while the network still remains under control of the operator. The approach provides the means for a dynamic system to automatically adapt to context changes. Moreover, an experimental system is presented for the purposes of concept validation and evaluation along with a real world use case.

## Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

## General Terms

Algorithms, management, reliability

## Keywords

LTE, management automation, policy, workflow

## 1. INTRODUCTION

Managing radio networks is a complex task especially in cellular mobile communication systems. The complexity is even increased if several radio networks should be seamlessly integrated. Complexity arises from the number of network elements (NEs) that have to be deployed and managed, but also from interdependencies between their configurations. In a heterogeneous network the variety of deployed technologies and their proprietary operational

paradigms are difficult to handle. Configuration, optimization, performance, and fault management require high expertise. Management tasks are typically performed by human operators and manual control is time-consuming, expensive, and error-prone.

Network management is usually based on a centralized operation, administration, and maintenance (OAM) architecture. Configuration and optimization of NEs is performed centrally from an operations and maintenance center (OMC) with support of a set of planning and optimization tools. However, this still requires a lot of human interaction. The staff uses their operational experience to find optimized configurations and they know when they have to diverge from standard procedures. Fault management procedures are also highly based on operational experience. Introducing new features through a software update or new NEs from different vendors requires changes in operational procedures.

Evolved Packet System (EPS) is the standard for next generation 3GPP mobile network evolution. Mobile network operators expressed a strong request for the introduction of self-organization features especially for this technology. Requests for operational expenditures (OPEX) reduction and for assuring high service quality call for changes in the way these networks are deployed and managed [1]. In order to provide high-quality services in the future, task automation is indispensable. Automation must go far beyond the possibilities used today such as scripting techniques. Although lots of tasks are executed identically, automation is difficult as fulfilling the same task in a heterogeneous network can require different actions on different NEs. Especially situations when only parts of NEs are updated increase the complexity.

To meet those challenges a new management approach is presented that reduces complexity for the operator by increasing automation. An efficient combination of workflows and policies enables the network with self-organizing capabilities while keeping a centralized management architecture. Although the basic architecture is centralized, parts of the functionality can be split and distributed over the NEs. However, some management functions such as alarm correlation will always require a central control entity. A major building block of the approach is the ability to execute workflows, another one is a policy framework to manage execution control. The presented approach is independent of the underlying radio technology but sufficiently generic to be applied to various network technologies and management domains.

The paper is structured as follows. A description of the basic technologies and concepts is given in Section 2. Section 3 presents the approach of policy-driven workflows. An experimental system is presented in Section 4 to demonstrate the results by means of a real-world use case, which is described in Section 5. The paper concludes with a summary and future work in Section 6.

© ACM 2010. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in:

*IWCMC '10*, June 28 - July 2, 2010, Caen, France.

## 2. BASICS

Workflows and policies are regarded as promising paradigms for network management. This section introduces them as an essential building block of the approach. The usage of both technologies in existing network management solutions is also illustrated.

### 2.1 Workflows

Workflows are broadly known from Business Process Management Systems or similar applications and have gained increasing interest from other areas. They describe a set of tasks that have to be performed sequentially or in parallel with the expected inputs and outputs. The definition used in this paper is based on [3]:

**DEFINITION 1 (WORKFLOW).** *Workflows orchestrate a set of activities to automate a business process and to accomplish a larger and more sophisticated goal according to a set of procedural roles.*

A variety of Business Process Management Systems and Workflow Systems exist, each one offering different features to a different extent [13], but no system dedicated to network management is known. Although the usage of Business Process Management and Service-Oriented Architecture has gained interest in the field of mobile communication, first approaches are only based on manually combined scripts.

Several powerful languages are available for modeling and executing workflows such as BPMN [4] and BPEL [5]. Such languages are typically XML-based, which allows validation against the syntactical description of the language. This simplifies verification of the syntactical correctness and debugging of workflows. Other systems feature a graphical interface for workflow specification. Resulting representations are subsequently parsed into the target workflow language.

Workflow languages (WFL) can be classified with workflow patterns, which refer to workflow language features such as splits and joins [7]. As soon as the available workflow patterns have been identified, it is possible to assess whether a given language is powerful enough to realize a workflow for a given problem. The basic strengths of a workflow language are:

- Formal workflow descriptions allow the validation of workflows at a high level of abstraction. Successful execution can be checked before deploying or executing workflows.
- A WFL allows to validate borders of possible inputs and outputs. This allows to specify inputs, but more important to check whether the enclosed activities can be combined according to the inputs.
- The possibility to replay the execution of a workflow step by step caters for easy debugging of the system.

Workflows have a dual appearance. On the one hand, they are considered as atomic parts which are executed as a single block; on the other hand, they can be constructed from a set of smaller atomic parts. These parts can either be single actions or workflows. Workflows can easily be adapted to a changing context by adding, removing, or exchanging parts. This duality promises to be a powerful means to meet the management challenges within the highly dynamical context changes of modern radio networks.

### 2.2 Policies

Policy-based management has gained attention in research and industry as a new paradigm as it allows operators to control and manage a system on a high level of abstraction. Policies are an appropriate means for modifying the behavior of a complex system without changing source code or considering its technical details [11]. A system can continuously be adjusted to externally imposed

constraints by changing the determining policies. A formal definition is [12]:

**DEFINITION 2 (POLICY).** *Policies are derived from management goals and define the desired behavior of a system. A policy is a set of declarative constraints on the possible behaviors of a system to define acceptable behaviors.*

The event-condition-action (ECA) model is a common way to specify policies. According to this model, a policy correlates a set of events, a set of conditions, and a set of actions to define which actions must or must not be performed in a certain situation. A situation is characterized by the conditions which define whether the policy is applicable or not and which are evaluated on the occurrence of an event. The actions are only executed if the conditions are met. Multiple policy frameworks share this model as for example Ponder2 [11].

A policy framework usually consists of a policy language, a policy engine, and a policy repository. Policies are represented either graphically or in a particular textual language. An execution engine enforces the policies by reacting to events and ensuring the actions are executed if the conditions are met. A policy repository is concerned with the storage aspect of policies. Policy frameworks typically provide means to cope with conflicts and prioritization in order to reduce complexity for the policy developer.

The usage of policy systems for management of mobile networks has already been considered in [2, 6] and there are also proposals for integrated management systems based upon this technology [9]. The possibility to easily change the system behavior is very appealing since major service interruptions are avoided. The usage of policies changes one very important aspect of network management. It allows tasks to be triggered from either the management console or from a self-organizing function within the network. Policies create a single interface to network management functions.

## 3. POLICY-DRIVEN WORKFLOWS

This section introduces the approach of policy-driven workflows for network management. It shows how operation and maintenance staff is freed from standard tasks and how policy-driven workflows allow radio network operators to automate tasks that are highly context-sensitive and therefore could not be automated previously.

### 3.1 Management Task Automation

Management tasks consist of a sequence of individual steps that are usually performed sequentially. These tasks can have interdependencies or depend on external circumstances. Recurring management tasks are identified and each management task is then mapped into a basic workflow. All steps of that management task are represented by steps within the workflow. An example workflow could be:

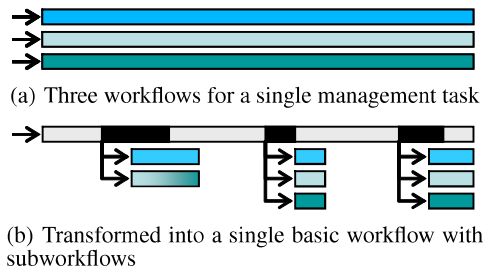
*New NE installed - determine radio technology - determine NE type - determine configuration - determine vendor and hardware - configure NE*

Although many occurrences of a management task are identical, there are numerous reasons that lead to smaller or larger deviations in their implementation. One reason is different APIs on equipment from different vendors or different parameter settings due to the outer context. In traditional network management knowledge and operational experience is used to cope with such issues. Implementing each of those variations into separate workflows brings two drawbacks:

- A large number of workflows would be necessary.
- A lot of workflows would have larger identical parts.

The approach follows a modular concept of workflows and subworkflows to overcome this issue. Any workflow can be reused as a part of other larger workflows and deviations between workflows can be swapped into separate subworkflows. Operators' experience is used to perform this separation. This leads to one single basic workflow per management task that dynamically includes the respective subworkflows for a particular situation as shown in Figure 1. Instead of having three separate workflows for a single management task, only one basic workflow is required. From the extracted subworkflows it can be seen that deviations from the basic workflow exist in three parts. In the first deviation the first workflow uses a separate subworkflow and the second and third workflow use a common subworkflow. In the second and third deviation all three workflows use separate subworkflows.

Separate subworkflows can be reused to configure hardware from different vendors. This addresses heterogeneity within the network but also allows to use the same workflows across different management applications for configuration, optimization, or fault management. Each workflow is only specified once, which avoids maintenance effort of redundant specifications and increases reuse and efficiency. For example, node insertion as well as node reconfiguration both require almost identical information about neighboring nodes and the sequence of configuration steps to be performed.



**Figure 1: Separation into Workflows and Subworkflows**

The selection of subworkflows in a particular situation is performed by the policy system and based on the current context. So-called selection policies are used for combining basic workflow and a subworkflows. Such policies are illustrated as angled arrows in Figure 1(b). They can use the vendor and hardware description to determine the correct configuration subworkflow. This provides a flexible way to update the functionality of the system or adapt it to changes in the network. If, for example, new hardware is introduced or the prerequisites in a region are changed, a new subworkflow is added and the respective policy is changed. Basic workflows are not touched and the system is not interrupted. There is no automated solution for separating task sequences into workflows and subworkflows yet. The separation is currently performed manually by the operator. Tool support that proposes a reasonable separation is future work.

Policies are also used initiate workflow execution. The execution of any workflow is triggered by events that contain any necessary contextual information. So-called trigger policies check the continuous event flow and decide to execute which workflow in which situation. They are ECA policies and are illustrated as direct arrows in Figures 1(a) and 1(b). The occurrence of an event signalizes the request for a decision to a respective trigger policy. Now, the policy condition is evaluated in order to determine whether to execute the workflow. This allows the integration of authorization

but also workflow coordination. This method is also used to delay workflow execution or to preempt running workflows when the requested workflow would invalidate the results of an already running workflow. A typical example is:

- **Event:** New NE inserted.
- **Condition:** No workflow running for this event AND event source authorized AND no higher priority workflow running.
- **Action:** Trigger workflow for initial NE configuration.

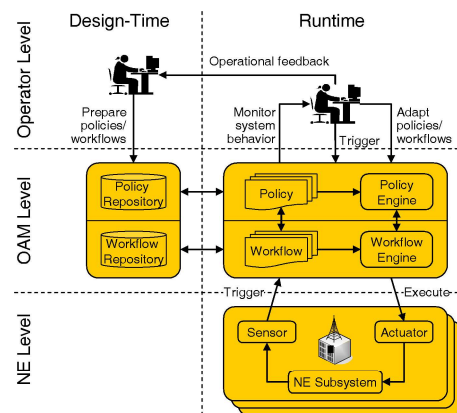
Workflows and policies have been used previously as techniques for business process management and network management, respectively, and there has been a lot of research in both areas. However, the combination of policies and workflows is a new approach to management automation and represents an effective way to increase automation whilst flexibility and control is kept for the operator at the same time. It is the first milestone towards operating a mobile network in a fully automated way. It is also the basis for integrating further techniques such as machine learning or probabilistic techniques.

### 3.2 Management Architecture

From an architectural point of view the combination of policies and workflows forms a middleware layer at the OAM level that integrates all necessary functionality for automated network management. Management tasks are executed without manual intervention whenever required. Triggering events may be issued from multiple sources in the network. An NE might autonomously request an initial configuration when newly inserted into the network. Workflow execution can also be triggered manually by an operator through a management application in case of manual updates or configuration changes. In both cases an event is generated, either by the management application or by the NE. This shows that the system is still under full control of the OAM staff. The remaining management tasks for operators are divided into tasks performed at design-time and tasks performed at run-time.

- **Design-time:** A pre-operational preparation phase for workflows and policies.
- **Run-time:** An operational phase where operators monitor the behavior of the network and derive adaptations of workflows and policies.

A fast control loop enables the operator to interact with the NEs. At runtime the operator monitors the system and changes its behavior by adapting policies or workflows. Experience gained at runtime is valuable feedback for the design of policies and workflows. Figure 2 summarizes the interaction between operator, middleware layer, and NEs at design-time and run-time.



**Figure 2: Architecture of the Management System**

The approach follows the divide-and-conquer paradigm as it separates functionality and control from each other. Workflows define execution logic and hence how things have to be done, whereas policies define decision logic and hence when and which workflows are executed. Separation of functionality and control increases the flexibility of management task execution. Fixed static workflows are not sufficient for use cases with interdependencies and concurrency. Policies react on a particular situation and resolve conflicts by the context-sensitive execution of workflows.

The difficulty of this separation is to determine an efficient distribution between fixed workflows and dynamic decisions. If the fixed workflows are too comprehensive, several redundant workflows for similar situations have to be provided. If the workflows are too brief, redundancy is minimized but the behavior of the system is difficult to handle due to excessive modularization. If there is a rather high number of workflows and policies, their overhead overrules their benefits. Finding the right degree highly depends on operational experience.

Keeping the number of simultaneously active policies at a low level also increases performance. In large scenarios the number of active policies could be very high. An incoming event causes the evaluation of the condition parts of all active policies, which delays the execution of workflows in case a lot of policies are evaluated that are actually not needed for a particular situation. To reduce the overhead, only the trigger policy for the basic workflow is active initially. All other policies are loaded when this policy triggers execution of the basic workflow. After the termination of the basic workflow they are disabled again. Thus, only policies that are currently required are active at any time.

#### 4. EXPERIMENTAL SYSTEM

Evaluation and validation of new management concepts is important in order to proof their applicability. For this reason, a highly modular experimental system has been created as a technical solution and a proof of concept for the combination of policies and workflows. It features all conceptual building blocks of the concept of policy-driven workflows. The system separates different functionalities into different modules as shown in Figure 3.

The modules are decoupled from each other and realize the different levels of the management system. They communicate through asynchronous events with a distributed publish-subscribe scheme. Alternatively, a synchronous communication mode allows for direct communication between two modules by sending an event and waiting for a dedicated reply event. Both communication modes are required for efficient workflow execution. Events are XML-based messages and may contain arbitrary information. Event types differentiate between different kinds of events. Modules can choose to receive all events or only events of a certain type. The implementation uses xmlBlaster [14] as message bus offering reliable event delivery, good integration with XML-based messages, and a larger number of different APIs.

The Knowledge module at NE level represents an information store for NEs and their configuration. Cells can be added sequentially or a complete network layout can be read from network planning files. Any NE can receive events to change its configuration or send events to e.g. request an initial configuration after insertion into the network. Orther modules can obtain information about NE configurations via the respective events.

At OAM level the Execution module realizes policy-driven workflows. It contains a policy and a workflow engine and offers functionality to specify and edit policies and workflows at design-time and run-time. Ponder2 is used as policy framework as it offers the functionality needed to control workflow execution using ECA

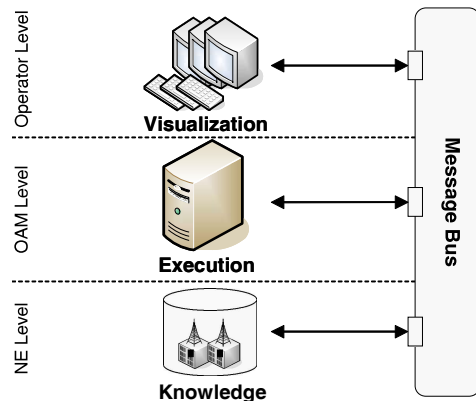


Figure 3: Architecture of the Experimental System

policies. Policies are stored within the Ponder2 framework and can be activated and deactivated. A simple Java-based workflow engine is implemented that offers good integration with Ponder2. Usage of a workflow engine that allows to formally specify workflows with a modeling language is planned and investigation for an appropriate engine is ongoing. The Execution module receives events to trigger policies and initiate workflow execution and sends events to interact with NEs or with the operator. A graphical user interface is available to monitor incoming and outgoing events. The control flow can be interrupted or slowed down. Events may also be queued to execute workflows step-by-step.

At operator level the Visualization module represents a management application that displays the current network layout and the configurations graphically on a map. It sends events to initiate workflows and receives all events that are relevant for the network configuration in order to always display the current configuration. Various instances of the Visualization module can be used at different locations while all instances always show the same information. Complex scenarios can be simulated, for example initiation of concurrent conflicting workflows.

#### 5. EXAMPLE: HARDWARE TO SITE MAPPING IN THE SELF-CONFIGURATION PROCESS

In order to validate and evaluate the approach with the experimental system, the important hardware to site mapping use case from the 3GPP mobile network domain was selected.

Deploying new radio base stations is a very complex and expensive task. After a long preparation phase with a lot of interactions between operator and vendors, the NEs are set up on-site where they will receive a location-specific configuration. The operator is informed about a certain hardware at a specific position, which enables him to continue with the configuration. The major disadvantage of this way of deploying a base station is that it is hardly possible to deploy hardware at a different place as initially intended. [8] provides a concept where an off-the-shelf base station is shipped without any preconfiguration. After installation and connection to power and data network, the location is determined and the auto-configuration process determines the configuration based on the location hardware ID.

The focus is now on the question how to determine the actual installation site which is transmitted to the OAM system together with the hardware ID to enable auto-provisioning. Determining the

site based on the geographical position with a very high probability is a challenging task for several reasons. Even with satellite-based positioning system such as GPS, for a small percentage of NEs the identification of a particular site is not possible. Especially co-located setups in urban areas are hard to handle as several NEs may be physically installed at the same site very close to each other. GPS measurement may deviate scores of meters in urban areas with a high building density. This mandates context-sensitive strategies for a correct hardware to site mapping.

Based on operational experience a basic workflow has been designed. Several methods to improve the quality of the site identification have been implemented as subworkflows. During run-time additional strategies based on new requirements can be specified and dynamically integrated. The following strategies are provided:

- **Closest:** Returns the site whose geolocation is closest to the measured position.
- **80:20:** Determines if one of the site is much closer than the other by taking into account the distances from the measured position to the two closest sites. The distances are compared to each other whether they satisfy the given ratio.
- **Active:** OAM databases record the state of NEs at a particular site. Sites which already have active NEs installed are excluded.
- **Type:** Site descriptions are extended with information on the expected NE type, e.g. macro, pico. All sites that are not intended for the given NE type are excluded.
- **MaxDistance:** All sites with a distance larger than some threshold value are excluded. This threshold has to be large enough to compensate measurement inaccuracies.

Depending on the context, one or any combination of these strategies are used to perform the site identification. Selection policies dynamically select the sequence of strategies to fit the context and the presets of the operator. Operator presets define a location-specific execution sequence and the threshold size. Each respective subworkflow reduces the set of possible sites. If at the end there is only a single site left, a successful mapping is assumed. If several candidate sites are left, operator input is required. This allows a fully automated hardware to site mapping in most cases.

## 6. CONCLUSION

A novel management approach was presented which combines workflow and policy technology into a management middleware layer to raise the state of self-organization. OAM staff is freed from time-consuming standard tasks, which allows them to focus on important tasks. In the best case a bare monitoring of the automatic functions and optimization of the operational concepts is sufficient.

To reach a state of operator-controlled autonomy, operational constraints and objectives are specified as policies and fixed task sequences are encoded into workflows. Policies control workflow execution and the adaptation of the control flow to the context. The usage of policies also allows to cope with workflow prioritization. Partitioning into workflows and subworkflows provides the means to cope with heterogeneity and network evolution without degradation in service quality.

The middleware layer decouples network management from the necessity of direct operator intervention and offers a modular way of organizing and reusing management functions. It also enables the autonomic functions to directly trigger management tasks, which closes the management loop. The approach is generic enough to be used within different management domains and parts can be used across multiple management domains.

However, it still raises a couple of challenges that need to be addressed in the future. Contradicting goals might occur between

concurrent workflows. Appropriate algorithms are needed to handle or at least detect such conflicts that cannot be handled by the policy framework. Additional use cases from different domains need to be implemented to evaluate the approach with quantitative measurements. Examples are alignment of radio parameters such as physical cell identifiers in the configuration management domain and cell outage compensation in the fault management domain. Furthermore, it is desired to use model-based techniques for the specification of workflows and policies which will offer more possibilities to analyze and verify the system behavior at design-time. This will even allow automated or at least semi-automated separation of tasks into basic workflows and subworkflows. In network management information models are already used today [10], but they are designed for specific management domains and cannot be used across different domains. The definition of policies and workflows should be taken to a more abstract level with a high-level language and a subsequent automated refinement into a machine-executable representation.

## 7. REFERENCES

- [1] H. Akhavan, T. Geitner, H. Dr. Lennertz, Y. Sha, T. Utano, and B. West. Next Generation Mobile Networks - Beyond HSPA & EVDO - Whitepaper. Technical report, NGMN Ltd., 2006.
- [2] T. Bandh, H. Sanneck, L.-C. Schmelz, and G. Carle. Automated Real-time Performance Management in Mobile Networks. In *IWAS 2007*. IEEE CS, June 2007.
- [3] T. H. Davenport and J. E. Short. The new industrial engineering: Information technology and business process redesign. *Sloan Management Review*, 31(4):11–27, 1990.
- [4] Object Management Group. BPMN Version 1.2. [www.omg.org/spec/BPMN/1.2](http://www.omg.org/spec/BPMN/1.2), January 2009.
- [5] Organization for the Advancement of Structured Information Standards. WS-BPEL Version 2.0. [www.omg.org/spec/BPMN/1.2](http://www.omg.org/spec/BPMN/1.2), April 2007.
- [6] R. Romeikat, B. Bauer, H. Sanneck, and C. Schmelz. A Policy-Based System for Network-Wide Configuration Management. In *18th WWRP*, June 2007.
- [7] N. Russell, A. H. ter Hofstede, W. M. P. van der Aalst, and N. Mulyar. Workflow Control-Flow Patterns: A Revised View. Technical report, BPMcenter.org, 2006.
- [8] H. Sanneck, C. Schmelz, T. Baumgarth, and K. Keutner. Network Element Auto-configuration in a Managed Network. In *IM '07*. IEEE CS, May 2007.
- [9] J. C. Strassner. *Policy-Based Network Management: Solutions for the Next Generation*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2003.
- [10] J. C. Strassner. Introduction to DEN-ng for PanLab II. [www.autonomic-communication.org/teaching/ais/slides/0809/Introduction\\_to\\_DEN-ng\\_for\\_PII.pdf](http://www.autonomic-communication.org/teaching/ais/slides/0809/Introduction_to_DEN-ng_for_PII.pdf), Jan 2009.
- [11] K. Twidle, E. Lupu, N. Dulay, and M. Sloman. Ponder2 - A Policy Environment for Autonomous Pervasive Systems. In *POLICY '08*, pages 245–246. IEEE Computer Society, 2008.
- [12] R. Wies. Policies in Network and Systems Management - Formal Definition and Architecture. *Journal of Network and Systems Management*, 2, 1994.
- [13] P. Wohed, N. Russell, A. H. Hofstede, B. Andersson, and W. M. Aalst. Open Source Workflow: A Viable Direction for BPM? In *CAiSE '08*, pages 583–586. Springer-Verlag, 2008.
- [14] xmlBlaster.org. xmlBlaster. [www.xmlBlaster.org](http://www.xmlBlaster.org), November 2008.