

Policies zum automatisierten technischen Netz-, System- und Anwendungsmanagement

Bewertung einer neuen Technologie
Policy-basiertes Management,
Management-Policies,
Policy-Steuerung

Forschungsbericht GFFT-2007-002

Schlüsselworte (Suchkriterien): Technisches Management, Management-Automatisierung,
Policy-basiertes Management, Management-Policies,
Autonomic Computing, Self-X-Techniken

Version 1.0 vom 13.9.2007

Geheimhaltungsgrad: Public

Autor: Heiko Krumm

Herausgeber: GFFT e.V.

GFFT e.V.
Taususstraße 23
61138 Niederdorfelden

INHALT

1	Executive Summary	3
2	Einleitung	4
3	Herausforderung.....	7
4	Einsatz der Technologie.....	7
5	Existierende Produkte	8
6	Empfehlungen	8
7	Kontakt	9

1 Executive Summary

Anwendungssysteme sollen sich selbsttätig an veränderte Randbedingungen anpassen können. Sie sollen sich flexibel auf neue Anforderungen einstellen und sich in dynamisch veränderliche Umgebungen einbetten können, z.B. auch problemlos mit neu hinzukommenden Anwendungen zusammenarbeiten.

Kommunikationsnetze, vernetzte Rechner und deren Betriebssoftware sollen den Anwendungen eine leistungsfähige und zuverlässige Plattform bilden. Sie sollen dazu automatisiert überwacht und administriert werden, so dass z.B. Ausfälle und beginnende Überlastsituationen schnell erkannt und automatisch behoben werden.

IBMs bekannte Initiative zum *Autonomic Computing* wie auch diverse Forschungsansätze zum so genannten *Organic Computing* fokussieren sich auf diese Zielsetzungen, die durch den Einsatz zahlreicher *Self-X*-Techniken erreicht werden sollen (i.e. Self-Configuring, Self-Healing, Self-Optimizing, Self-Protecting, Self-Organization, Self-Regulation, Self-Repair, Self-Maintenance and Self-Diagnosis). Die derzeit diskutierten Verfahren bilden ein breites und vielfältiges wie auch spekulatives Feld. Einzelne Verfahren werden sich in der Praxis bewähren, andere nicht.

Bei wohl den meisten der für den praktischen Einsatz geeigneten Automatisierungsansätze spielen *Technische Management Policies* eine zentrale Rolle:

- Abstrakte – High-Level – Policies definieren Zielvorgaben, Richtlinien und Rahmenbedingungen. Sie entsprechen oft den Sollwerten in einem klassischen Automatisierungssystem.
- Detaillierte, Technik-nahe – Low-Level – Policies haben den Charakter von Handlungsanweisungen, Regeln und konkreten Betriebsbedingungen. Sie entsprechen oft den Regelalgorithmen des klassischen Automatisierungssystems.

Policies werden auf der Basis von Policy-Modellen und Policy-Sprachen (z.B. Web Services Policy: *WS-Policy*) – oft mit Hilfe eines Policy-Editors (z.B. *Microsoft Policy Editor*) – erstellt. Interessant ist es, wenn nur abstrakte und übersichtliche Policies editiert werden und daraus mechanisch adäquate Low-Level-Policies generiert werden. Diese werden im System an Policy-Enforcement-Komponenten verteilt und von ihnen zur Laufzeit ausgewertet und durchgesetzt. Sehr bekannte aber funktionell eingeschränkte Beispiele dazu sind Netzwerk-Firewalls, deren Filterlisten direkt interpretierbare Low-Level-Policies darstellen. Ein umfassenderes System ist das von *IBM/Tivoli* angebotene *PMAC* (Policy Management for Autonomic Computing).

Policies wurden zunächst hauptsächlich im Sicherheitsbereich, insbesondere als einseitige, systembezogene Zugriffskontroll-Regeln und als Filterregeln, eingesetzt. Auch bei Web Services konzentriert sich ihr Einsatz noch auf die Sicherheitsanforderungen von Client und Server. Allerdings haben hier beide Parteien je eine eigene Policy, und zur Laufzeit wird eine Auswahl aus der zulässigen Schnittmenge getroffen. Es werden also Policies unterschiedlicher Parteien miteinander abgestimmt, um die Modalitäten einer Kooperation festzulegen.

Weitergehende Anwendungen von Policies finden sich in der Softwaretechnik. Programme mit Anpassungsfähigkeit werden so gestaltet, dass sie zur Laufzeit die aktuell gültigen Policies erfragen und anhängige Entscheidungen an Hand dieser Policies treffen können. Policies bilden so oberhalb des Programmcodes eine neue, dynamisch veränderliche Steuerungsebene, welche per Administration oder per automatischer Abstimmung mit den Policies der Umgebung flexibel und kurzfristig an aktuelle Anforderungen angepasst werden kann.

Technische Policies stellen wegen dieser Verwendungsmöglichkeiten ein immer wichtiger werdendes Hilfsmittel zur Gestaltung vernetzter IT-Systeme dar, das in seinem Potential gegenwärtig von vielen aber noch unterschätzt wird, das sich aber in naher Zukunft als zentrales Lösungselement zur Bildung flexibler, adaptiver IT-Systeme erweisen wird. Wir empfehlen

deshalb, schon jetzt bei der Produktauswahl auf Policy-Steuerungsmöglichkeiten zu achten und die systematische Erfassung der bestehenden Richtlinien und Zielvorstellungen vorzubereiten.

2 Einleitung

Der Policy-Begriff ist auch im engeren Feld des technischen Managements und der adaptiven Steuerung von Anwendungssystemen sehr breit gefasst und – wie das umgangssprachliche Wort „Policy“ – schwer eindeutig definierbar. Wir geben deshalb eine Übersicht über das Spektrum anhand der wichtigsten Policy-Formen des technischen IT-Managements.

Klassische Management-Policies

Technische Management Policies sind diejenigen Richtlinien, nach welchen die Netz-, System- und Anwendungsverwaltung eines IT-Systems durchgeführt werden soll. Sie werden im klassischen Ansatz in einem kreativen Prozess aus den unternehmerischen Zielen unter Beachtung der technischen Zwänge abgeleitet und informell als „*Management Policy*“ notiert. Die (menschlichen) Administratoren kennen und beachten die Policies während ihrer Arbeit.

Automatisierung durch Policy-based Management

Maschinell verarbeitbare Policies unterstützen in diesem Ansatz die Automatisierung der technischen IT-Verwaltung:

- Die Policies werden in einer formalen Policy-Sprache (oder in ihrer Syntax festgelegten Tabellen) formuliert, ihre Semantik ist über einem Policy-Modell definiert.
- Die Policies werden mit einem Policy-Editor erstellt und an verschiedene Policy-Enforcement-Komponenten im System verteilt.
- Die Policy-Enforcement-Komponenten überwachen den laufenden Betrieb, prüfen ihn auf Verträglichkeit mit den Policies und greifen u.U. wiederum gesteuert durch Policies korrigierend ein.

Ein sehr bekannter entsprechender Ansatz ist der Internet-Policy-Ansatz der IETF. Das „Policy Core Information Model“ ist ein objektorientiertes Modell, das Grundlage für die Definition von Policies bildet. Policies bestehen aus Regeln. Jede Regel ist ein Paar aus Bedingung und Aktion. Die Enforcement-Komponenten prüfen die Bedingungen zugeordneter Policy-Regeln und feuern gegebenenfalls die verbundenen Aktionen.

Einfache Policies

In einfachen Anwendungen des Policy-based Managements sind die Bedingungen und Regeln der Policies direkt an technischen Systemparametern orientiert. Ein typisches Beispiel dazu ist die Bestimmung der Priorität eines IP-Pakets anhand von Tageszeit, Quell-, Zieladresse und Protokollkennung. Ein anderes häufiges Beispiel ist die Konfiguration der *IPsec*-Implementierung eines Betriebssystems (z.B. bei Linux oder bei Microsoft Windows) anhand der *IPsec*-Policy-Definition. Sie legt fest, wie IP-Pakete in Abhängigkeit zu ihren Quell- und Zieladressen gesichert zu übertragen sind. Ein weiteres Beispiel ist CISCOs Policy-based Routing (PBR). Hier bestimmen einfache Regeln, über welche Internet-Anbindung abgehende Pakete geleitet werden. Die meisten aktuellen Produkte mit Policy-based Management unterstützen ausschließlich Low-Level-Policies eines sehr eingeschränkten und produktspezifischen Anwendungsbereichs.

Integriertes Management mit übergreifenden Policies

Gegenwärtig ist aber eine Phase der Integration erkennbar. Übergreifende Policies berücksichtigen umfassende, für das Gesamtsystem geltende Zielsetzungen. Sie werden im gemeinsamen Zusammenhang modelliert, erstellt und auf wechselseitige Verträglichkeit geprüft. Per Tool werden daraus dann erst die für die einzelnen Enforcement-Komponenten geeigneten Policy-Teilmengen gewonnen.

Policy-Hierarchien

Übergreifende Policies werden vorteilhaft auf einem höheren Abstraktionsniveau definiert, in welchem Details konkreter Systemkomponenten verborgen bleiben. Die sehr abstrakten Policies der höchsten Ebene werden mit der Firmenleitung abgestimmt. Sie werden schrittweise über mehrere Stufen verfeinert und detailliert, so dass letztendlich automatisch durchsetzbare Low-Level-Policies gewonnen werden, welche die Einhaltung der abstrakten Policies gewährleisten. Die Policy-Verfeinerung ist i.a. nicht allein maschinell durchführbar, kann aber inzwischen systematisch und mit Tool-Unterstützung geleistet werden, so dass Aufwand und Fehleranfälligkeit des Policy-Übersetzungsprozesses begrenzt bleiben.

Dieser „Policy-Hierarchien“ genannte Ansatz ist zwar bis jetzt noch nicht in gängigen Produkten zu finden, weist aber das Potential auf, heterogene vernetzte IT-Systeme effizient und einheitlich zu verwalten.

Policies zur System- und Anwendungssteuerung

Traditionellerweise werden die hier betrachteten technischen Policies zur direkten Steuerung von Administrationsfunktionen (Überwachung, Konfiguration, Fehlerdiagnose, Reparatur) verwendet, also von spezieller Management-Software ausgewertet. Sie spielen aber inzwischen auch bei genereller Anwendungs- und Systemsoftware eine wachsende Rolle, denn moderne Systeme (egal ob Anwendungen, Server-Software, Betriebssysteme oder Middleware), welche Selbstadaption durchführen, bewirken dies in der Regel dadurch, dass sie programmgesteuert selbstständig Administrationsfunktionen auswählen, parametrisieren und aufrufen. Hier ergibt sich ein neues und sehr bedeutendes Aufgabenfeld für Policies. Sie bilden eine zusätzliche, oberhalb des Programmcodes liegende Steuerungsebene, welche nicht mit dem Programmcode festgelegt ist, sondern welche dynamisch zur Laufzeit verändert werden kann. Bei zukünftigen, aus selbstverwaltenden Komponenten bestehenden IT-Systemen finden sich dann im Groben folgende 4 Ebenen der Betriebssteuerung:

1. Das technische Management pflegt die *Policies in ihrer Gesamtheit* und verteilt sie an die Komponenten. Neben ausgesprochenen Management-Komponenten werden auch Anwendungen, Services und Betriebssysteme mit Policies versorgt. Die Gesamt-Policy wird vorteilhaft entsprechend zum Ansatz der Policy-Hierarchien auf einer höheren Abstraktionsebene formuliert.
2. Die einer *Komponente aktuell zugeordneten Policies* beschreiben die momentan gültigen Randbedingungen und Zielsetzungen des Betriebs dieser Komponenten.
3. Die Komponente enthält Code, welcher zur Laufzeit *Entscheidungen an Hand der aktuell gültigen Policies* trifft.
4. Der Hauptanteil des Codes der Komponenten implementiert – wie bisher auch – die *eigentliche Komponentenfunktionalität*. Sie hat jedoch nun Varianten und Parameter, welche durch Policies beeinflussbar sind.

Die Nutzung von Policies und die entsprechenden 4 Steuerungsebenen sind inzwischen hinsichtlich des Aspekts Security schon selbstverständlich geworden. Die Plattformen CORBA und Java spiegeln das z.B. sehr direkt wieder. Policies werden dabei zur Laufzeit als Policy-Objektinstanzen in ein Programm eingebracht, deren Methoden Policy-basierte Entscheidungen treffen. Die Entscheidungen konzentrieren sich dabei auf das Gewähren oder Ablehnen von Ressourcen-Zugriffen.

Eine Ausweitung der Policy-Steuerung von Anwendungen auf andere Felder neben der Security ist absehbar, und z.B. schon im Produkt PMAC (Policy Management for Autonomic Computing) von IBM/Tivoli realisiert.

Policy-Felder

Die prognostizierte Ausweitung der Policy-Steuerung findet durch Erschließen neuer Aufgabenfelder statt, die über das klassische Feld der Anwendungssecurity hinausgehen. Die Felder lassen sich 2-dimensional anordnen. Eine Richtung bilden die *Problembereiche*, wie z.B.:

- Sicherheit,
- Leistung,
- Fehlerdiagnose und Fehlertoleranz,
- Kommunikation,
- Abrechnung,
- Partnerauswahl und Bindung.

In der zweiten Richtung sind die von einer Policy betroffenen Parteien unterscheidbar, wie z.B.:

- Nutzer,
- Gerät,
- Betriebssystem,
- Anwendung,
- Firma,
- Kooperationsbeziehung.

Innerhalb desselben Systems kann es also z.B. gleichzeitig mehrere gültige Security-Policies geben. Die einem Nutzer zugeordnete Policy beschreibt z.B. seine Rechte, Pflichten und Anforderungen, und die einer Anwendung zugeordnete Policy beschreibt die generell für den Betrieb der Anwendung geltenden Regeln. Wenn nun ein bestimmter Nutzer mit einer bestimmten Anwendung arbeiten will, müssen beide Policies in Einklang gebracht werden und der Betrieb muss die Bedingungen beider Policies erfüllen.

Policy-Abstimmung

In SOA-Umgebungen werden insbesondere Abstimmungen zwischen Client- und Server-Policies notwendig, die einem Aushandeln der aktuellen Nutzungsbedingungen entsprechen. In zukünftigen „Pervasive Computing“-Umgebungen werden darüber hinaus eine Vielzahl weiterer Abstimmungen benötigt, um sicherzustellen, dass hinzukommende Geräte sich in eine bestehende Umgebung reibungslos integrieren können, und die Interessen der verschiedenen Parteien (Nutzer, Provider, Eigentümer) gewahrt werden. Basisfunktion dazu ist die Berechnung der Schnittmenge (logisch: der UND-Verknüpfung) einer Menge von Policies, weil die Schnittmenge den Spielraum darstellt, welcher bei Beachtung aller Parteien-Policies übrig bleibt. Entsprechende Erweiterungen des WS-Policy-Ansatzes sind in Arbeit, u.a. wird von der Fa. *Sun* eine XACML-basierte Policy-Darstellung mit effizient berechenbarer Schnittmenge vorgeschlagen.

Policy Modelle und Semantik

Syntax und Semantik von Low-Level-Policies sind – wie schon erwähnt – in der Regel direkt an der Funktion und den Konfigurationsparametern einer bestimmten Systemkomponente ausgerichtet. Auf höheren Abstraktionsebenen besteht dagegen der Wunsch nach Implementierungsunabhängigkeit und es ist Ziel aktueller Forschungs- und Entwicklungsarbeiten, abstraktere Semantik-Definitionen zu unterstützen, welche nicht letztendlich zur nicht-formalen Semantik-Definition per natürlich-sprachlichem Kommentar Zuflucht nehmen.

Ein Ansatz dazu – der z.B. von der DMTF mit dem CIM Policy Model verfolgt wird – sind objektorientierte Systemmodelle, welche auf der Basis vordefinierter genereller Komponentenklassen gebildet werden. Konkrete Komponenten einer Systemimplementierung instanzieren Klassen, welche Verfeinerungen der generell im Standard definierten Klassen sind, und die Bedeutung der Policies wird damit durch den Bezug auf die Standard-Klassen festgelegt.

Eine weitere, aktuell an vielen Stellen verfolgte Stoßrichtung ist es, sich die im Rahmen von Web 2.0 entstehenden Ontologien zu Nutzen zu machen, um offene, leicht erweiterbare und leicht mit neuen Funktionsfeldern verknüpfbare Definitionen zu ermöglichen.

3 Herausforderung

Gegenwärtiger Stand der Technik ist, dass schon viele Produkte mit einer Policy-Steuerung versehen sind. Es handelt sich dabei in der Regel um Low-Level-Policies, welche direkt am einzelnen Produkt ausgerichtet sind. In den klassischen Policy-Anwendungsfeldern Security und ID-Management existieren darüber hinaus Produktansätze mit übergreifenden Policies.

Da mindestens mittelfristig Service-orientierte Architekturen mit flexiblen, dynamisch zur Laufzeit erweiterbaren und anpassbaren IT-System-Strukturen realisiert werden müssen, ist vorhersehbar, dass Policy-Steuerungen weiter Raum greifen.

In der gegenwärtigen Situation ist es vor diesem Hintergrund wichtig, heute schon die Voraussetzungen zu schaffen, um später übergreifend automatisierbare Gesamtlösungen zu ermöglichen.

4 Einsatz der Technologie

Voraussetzung für den Einsatz der Policy-Steuerung auf Komponenten-Ebene ist, dass die einzelnen Komponenten für sich gesehen mit Policy-Steuerungsmöglichkeiten versehen sind.

Voraussetzung für die integrierte, übergreifende und systematische Handhabung der Policies in ihrer Gesamtheit ist der Entwurf eines entsprechenden Konzepts. Es soll eine abstrakte, übergreifende High-Level-Policy, eine davon ausgehende Hierarchie, welche letztlich alle vorhandenen Low-Level-Komponenten-Policies erreicht, sowie Regeln zur Ableitung der Low-Level-Policies per Verfeinerung umfassen.

Zur Implementierung der hier empfohlenen übergreifenden Policy-Behandlung sind hauptsächlich Identifikations-, Planungs- und Entwurfsarbeiten zu leisten:

- Sammlung der verschiedenen bestehenden Zielvorstellungen, Richtlinien und Regeln,
- Sortieren und Strukturieren der Richtlinien im Sinne einer abstrakten übergreifenden Policy,
- Erfassung der vorkommenden Policy-gesteuerten Komponenten und Low-Level-Policies,
- Entwurf einer Policy-Hierarchie, welche abstrakte, übergreifende Policy und Low-Level-Policies verbindet,
- Entwurf der (zunächst manuell anzuwendenden) Prüf- und Ableitungsprinzipien, mit welchen die wechselseitige Verträglichkeit von Teil-Policies geprüft und Komponenten-Policies aus der abstrakten Policy abgeleitet werden,
- Dokumentation der Entwurfsergebnisse insbesondere in Form eines an IT-Administratoren gerichteten Handbuchs zur Policy-Behandlung in der Unternehmens-IT.

Besondere Aufwände und Kosten entstehen weniger durch die Wahl Policy-steuerbarer Komponenten, als vielmehr durch den Personalaufwand zur Planung und zum Entwurf der übergreifenden Policy-Behandlung. Je nach dem Umfang des IT-Systems, den vorhandenen Policy-gesteuerten Komponenten sowie der Breite der durch die Policy-Steuerung erfassten Aspekte ist mit einem Personalaufwand zwischen 6 und 24 Personenmonaten zu rechnen.

Direkte besondere Risiken entstehen durch den Einsatz Policy-gesteuerter Produkte nicht. Die Erarbeitung der übergreifenden Policy birgt als Komplement zum Potential effizienzsteigernder Klärung auch das Risiko nicht-adäquater Policy-Gestaltung und daraus resultierender Erschwernisse des IT-Betriebs.

5 Existierende Produkte

Wie oben erwähnt gibt es inzwischen eine Vielzahl Policy-steuerbarer Einzelprodukte. Produkte zur integrierten Definition, Pflege und Umsetzung übergreifender Policies in der Breite sind noch nicht vorhanden.

Viele Produkte sind dem Bereich des Policy-basierten Netzmanagements zuzuordnen. *Lucent* wendet hier z.B. die Policy Sprache PDL im Produkt *RealNet Rules Policy Management Application* an. *Cisco* implementiert explizit Policy-basierte Mechanismen zur Dienstgüte und zur Routing-Steuerung. Es ergibt sich hier eine enge Querbeziehung zwischen Policies und Skripten. Das Netz- und Anwendungsmonitoring-System *BMC Patrol* verwendet ebenfalls (als so genannte Knowledge Module) Skript-Prozeduren, die Funktionen von Low-Level-Policies wahrnehmen. Produkte, welche die IP-Sicherheitserweiterungen *IPsec* implementieren (z.B. Linux, Windows, VPN-Konzentrator-Produkte), besitzen nach Standard eine Policy-Management-Schnittstelle.

Im Aufgabenfeld des System- und Anwendungsmanagements haben sich insbesondere im Umfeld der *Microsoft Windows Server* Betriebssysteme Policy-basierte Verfahren durchgesetzt. Es gibt allerdings nicht einen einzigen integrierenden zentralen Policy-Ansatz, sondern es werden verschiedene Policy-Ansätze – jeweils auf einen engeren Zweck bezogen – verwendet. Die zu Grunde liegenden Policy-Modelle sind in der Regel sehr einfach gehalten. Policy-Elemente entsprechen – wie bei den *IPsec*-Policies der *IETF* – direkt Konfigurationsparameter-Datensätzen.

Beispiele für enger eingegrenzte Policy-Ansätze im Umfeld der *Microsoft Windows Server* Betriebssysteme sind die *Konfiguration der Sicherheitsmechanismen* per *IPsec*-Policies, die *Group Policy Management Console (GPMC)* bei *Windows Server 2003* bzw. der *Group Policy Editor (GPE)* bei *Windows XP Professional* sowie die *Fehlermeldungssteuerung* im *Corporate Error Reporting (CER)* System, bei welchem sich je Applikation Policies definieren lassen, um zu steuern, ob und welche Fehlermeldungen erzeugt und wie sie weitergegeben werden.

Als neueres und gegenwärtig stark in's Bewusstsein tretende Aufgabenfeld erweist sich das ID-Management, für welches ebenfalls schon Policy-basierte Produkte verfügbar sind, z.B. bietet Sun ein integriertes Policy- und ID-Management als Bestandteil des *iPlanet Directory Server* Produkts an.

6 Empfehlungen

Um den zukünftigen Anforderungen heute schon entgegenzukommen, sollten folgende Maßnahmen durchgeführt werden:

- Der Entwurf eines abstrakten übergreifenden Policy-Gesamtkonzepts, das erweiterbar und hierarchisch angelegt ist, dient dazu, die Management- und Adaptionanforderungen des Unternehmens in einer dokumentier- und prüfbarer Weise zu erfassen.
- Bei Neubeschaffungen sollte auf Policy-Steuerungsmöglichkeiten geachtet werden. Es sollte berücksichtigt werden, wie weit die Policy-Steuerung in das Gesamtkonzept passt.

- Für alle im IT-System vorhandenen Komponenten, die eine Policy-Steuerung besitzen, sollte der Bezug zum Policy-Gesamtkonzept hergestellt werden, d.h. es sollte jeweils definiert werden, wie die konkrete Komponenten-Policy per Policy-Verfeinerung aus der abstrakten Gesamtpolicy abzuleiten ist.

Die Maßnahmen sollen gewährleisten, dass die nach und nach im IT-System aufkommende Policy-Welt aus harmonisierbaren Komponenten besteht, um die Policy-Durchsetzung mittelfristig mit hohem Automatisierungsgrad realisieren zu können.

Als unmittelbarer Vorteil der systematischen Erfassung, Dokumentation und Modellierung der bestehenden Policies ergibt sich, dass Inkonsistenzen und Zielkonflikte des IT-Betriebs frühzeitig entdeckt und aufgelöst werden können.

7 Kontakt

Prof. Dr. Heiko Krumm
FB Informatik, LS4, FG RvS
Universität
44221 Dortmund

Tel.: 0231-7554674
Fax: 0231-7554730
E-Mail: Heiko.Krumm@udo.edu
WWW: <http://ls4-www.informatik.uni-dortmund.de/RVS/agrvs.html>