



AW1-Vortrag

Autonomic Computing (AC)

Maik Weindorf

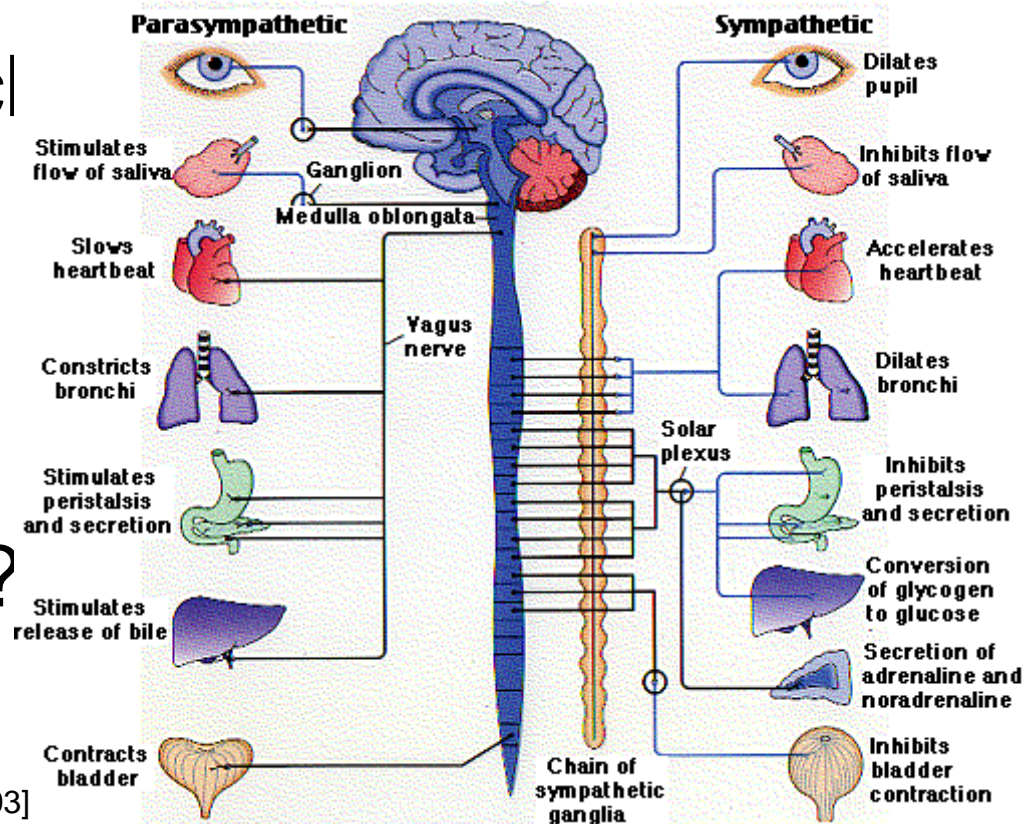
Hamburg, 22. Juni 2006

- Motivation
- Grundlagen
- Architektur
- Produkte
- Fazit/Ausblick
- Literatur

- Motivation
- Grundlagen
- Architektur
- Produkte
- Fazit/Ausblick
- Literatur

Begriff und Grundidee

- Analogie/Metapher zum „Autonomen Nervensystem“ (bzw. „Vegetativen Nervensystem“)
 - Anpassung an versch
 - Ohne externe Hilfe
- Wunschzustand für Computersysteme (?)





Problemstellung

- Wunsch nach Kontrolle und Steuerung von Geschäftsprozessen (SLAs, KPIs)
- Häufige Änderungen von Geschäftsprozessen (vgl. Workflow-Management, BPEL, SOA, ...)
- Wachsende Vernetzung von Systemen
- Heterogenität von Computersystemen

Folgen:

- Hohe und stark wachsende **Komplexität** von Computersystemen (→ manuell nicht beherrschbar)
- Hohe **Kosten** für Change-Management, Wartung, Problemerkennung und -behebung



Lösungsansätze

- UC Berkeley/Stanford: Recovery-Oriented Computing
 - Microreboots und Crash-Only Software
- Carnegie Mellon University: Architecture Based Languages and Environments
 - Project Rainbow: Architecture-based Adaptation of Complex Systems
- „Autonome“ Eigenschaften von Teilsystemen (Applikationen, Server, ...)
 - z.B. Datenbanken (Load-Balancer, Recovery, etc.)
- **IBM: Autonomic Computing**

Historie

- 2001: IBM startet „Autonomic-Computing-Initiative“ (+ „Autonomic Computing Manifesto“)
- 2003: Autonomic Computing Workshop auf dem 5th Annual International Workshop on Active Middleware Services (AMS 2003)
- 2004: 1st IEEE International Conference on Autonomic Computing (ICAC 2004) → seitdem jährlich

- Motivation
- **Grundlagen**
- Architektur
- Produkte
- Fazit/Ausblick
- Literatur



Anforderungen (AC System)

Ein AC System:

- muss sich selbst „kennen“
- muss sich selbst konfigurieren
- erreicht nie einen „Status Quo“
- muss sich selbst heilen
- muss sich selbst schützen
- muss seine Umgebung kennen
- muss in einer heterogenen Umgebung funktionieren (→ kann keine proprietäre Lösung sein)
- muss den Benutzern gegenüber seine Komplexität verbergen

AC “functional areas”

Self-CHOP:

Increased Responsiveness

“Adapt to dynamically changing environments”

Operational Efficiency

“Tune resources and balance workloads to maximize use of IT resources”



Business Resiliency

“Discover, diagnose, and act to prevent disruptions”

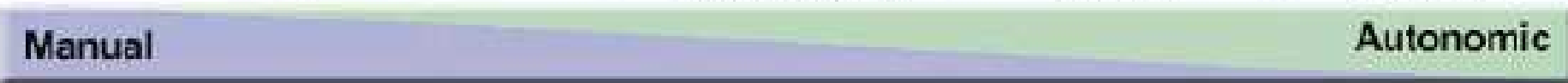
Secure Information and Resources

“Anticipate, detect, identify, and protect against attacks”

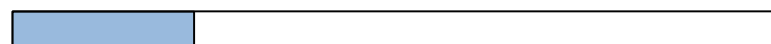
[ibm-04]



Levels of Autonomic Computing Maturity

	Basic Level 1	Managed Level 2	Predictive Level 3	Adaptive Level 4	Autonomic Level 5
Characteristics	Rely on system reports, product documentation, and manual actions to configure, optimize, heal and protect individual IT components	Management software in place to provide consolidation, facilitation and automation of IT tasks	Individual IT components and systems able to monitor, correlate and analyze the environment and recommend actions	IT components, individually and collectively, able to monitor, correlate, analyze and take action with minimal human intervention	Integrated IT components are collectively and dynamically managed by business rules and policies
Skills	Requires extensive, highly skilled IT staff	IT staff analyzes and takes actions	IT staff approves and initiates actions	IT staff manages performance against SLAs	IT staff focuses on enabling business needs
Benefits	Basic requirements addressed	Greater system awareness Improved productivity	Reduced dependency on deep skills Faster/better decision making	Balanced human/system interaction IT agility and resiliency	Business policy drives IT management Business agility and resiliency
Manual  Autonomic					

[ibmred-05]

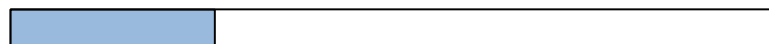




Levels of Autonomic Computing Maturity

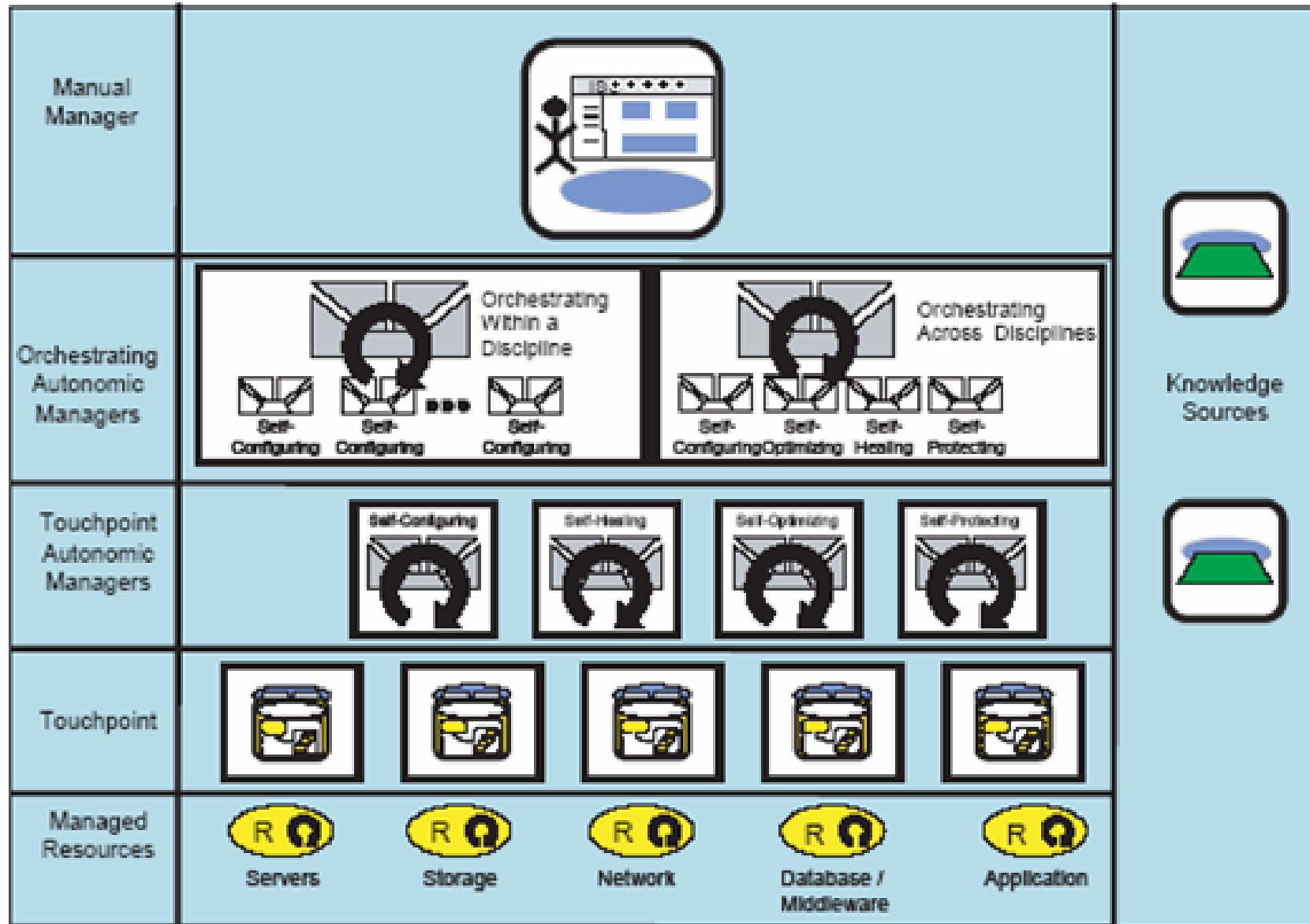
Zusammengefasst:

- **1 Basic** manuelle Analyse und Problembehebung
- **2 Managed** zentralisierte Tools, manuelle Maßnahmen
- **3 Predictive** „Cross-Resource“ Korrelation und Führung
- **4 Adaptive** „System Monitors“ erkennen Korrelationen und führen Maßnahmen aus
- **5 Autonomic** dynamisches Management auf Basis von Geschäftsregeln



- Motivation
- Grundlagen
- **Architektur**
- Produkte
- Fazit/Ausblick
- Literatur

Architektur Überblick



[ibm-05]

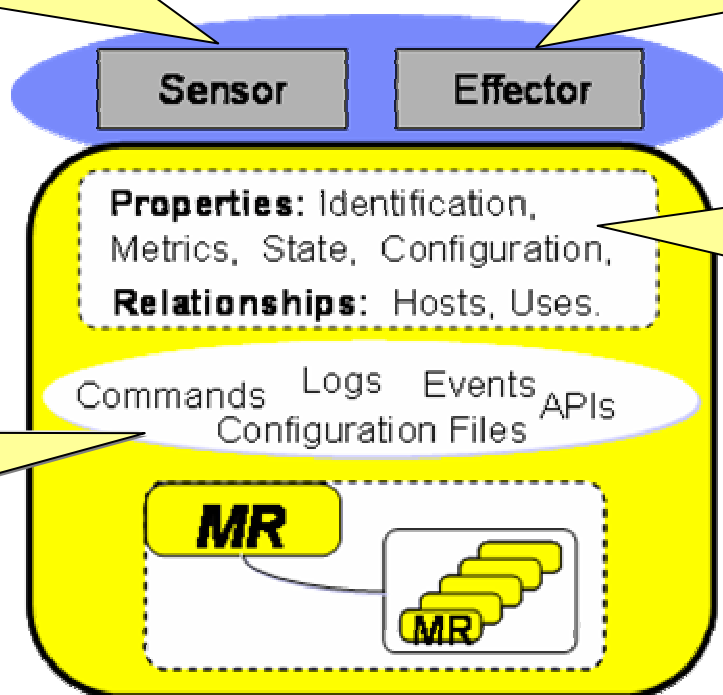
Touchpoint

“A touchpoint is an autonomic computing system building block that implements the sensor/effector pattern for one or more of the manageability interface mechanisms.”

Manager erfahren Details durch den **Sensor**.
Zwei Verfahren:
(1) Manager fordert Details an
(2) Resource bietet Details an

Manager ändern Details durch den **Effector**.
Zwei Verfahren:
(1) Manager initiiert die Änderung
(2) Resource fordert die Änderung an

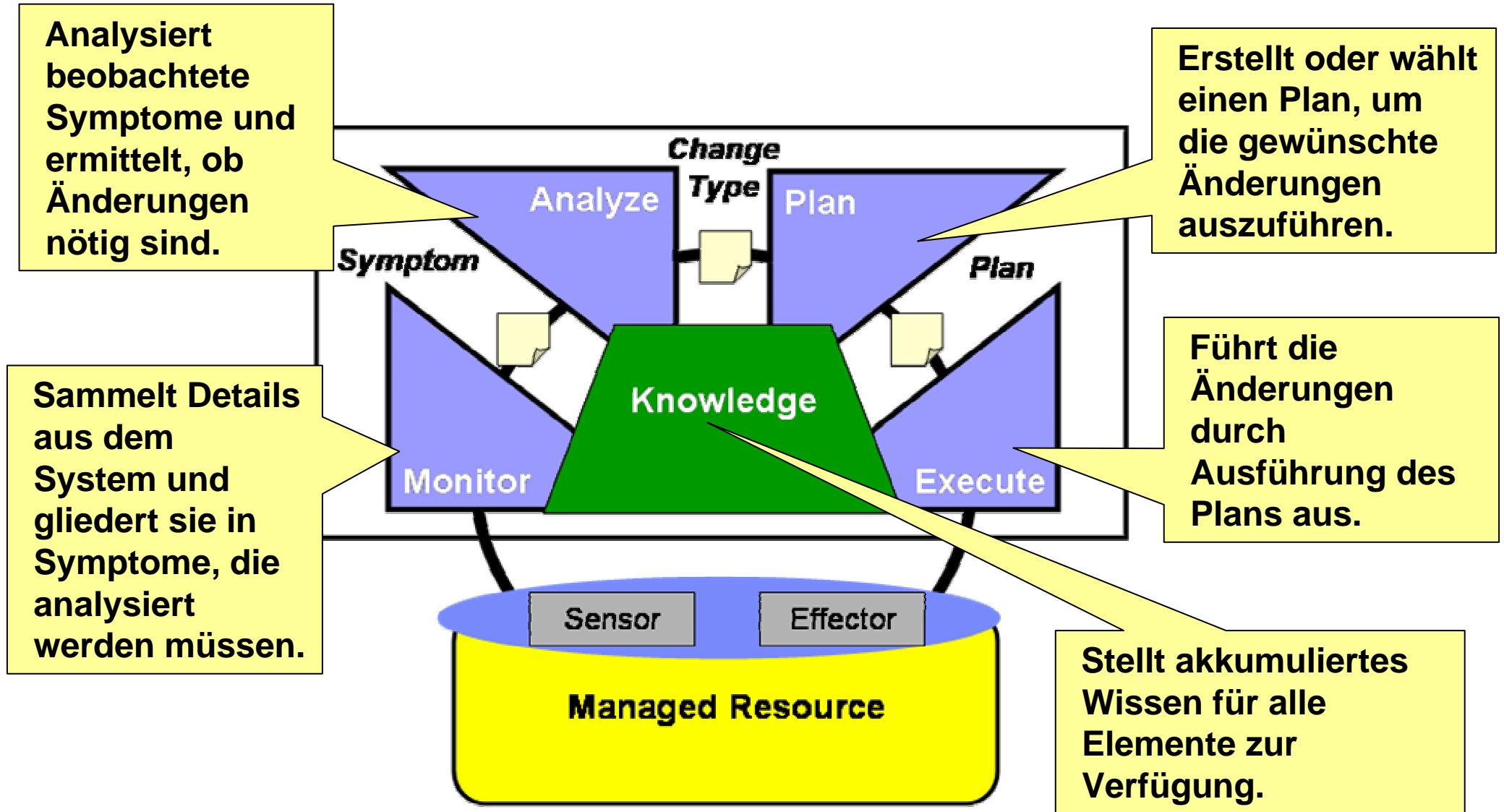
Die Mechanismen um eine „Managed Resource“ zu überwachen und zu steuern.
(Manageability Interface)



Die Details, die ein Monitor benötigt, um eine „Managed Resource“ zu überwachen und zu steuern.

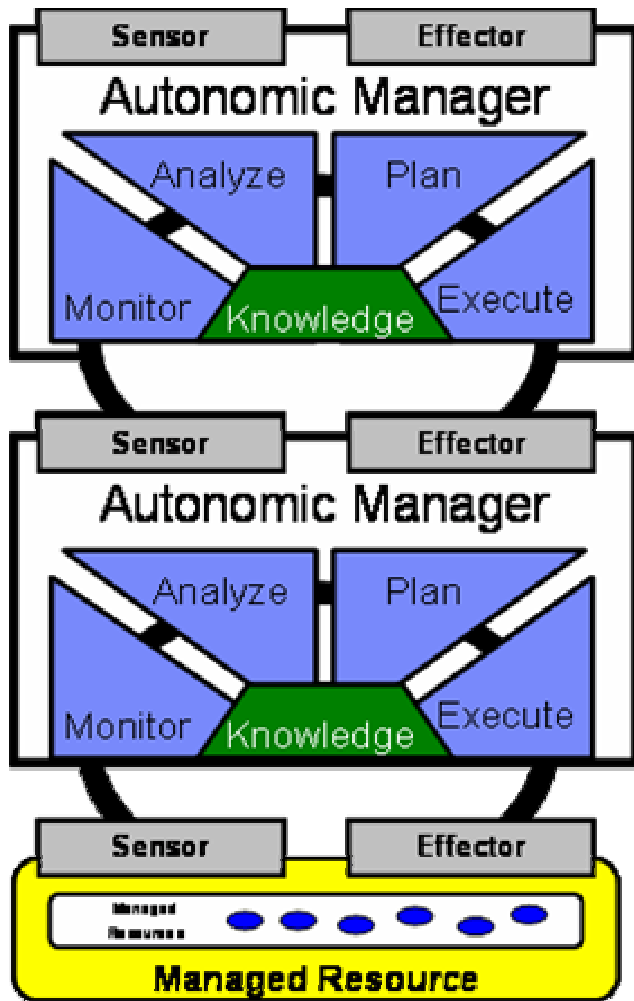
[ibm-04]

Autonomic Manager



[ibm-04]

Autonomic Manager



[ibm-04]

“Orchestrating” AUTONOMIC MANAGER

- Nimmt höhere Geschäftsziele entgegen
- Übersetzt Geschäftsregeln in Ziele und Richtwerte für die von ihm verwalteten Ressourcen
- Gibt Ziele weiter (an Ressourcen)

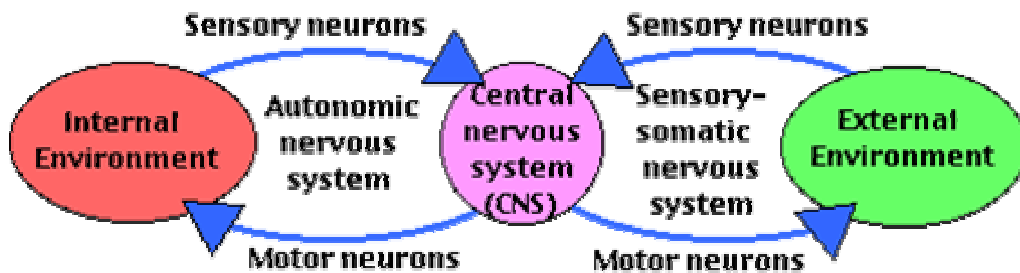
“Touchpoint” AUTONOMIC MANAGER

- Nimmt Ziele entgegen
- Übersetzt Ziele (→ Effectoren)
- Steuert über Effectoren und misst Ziele über Sensoren

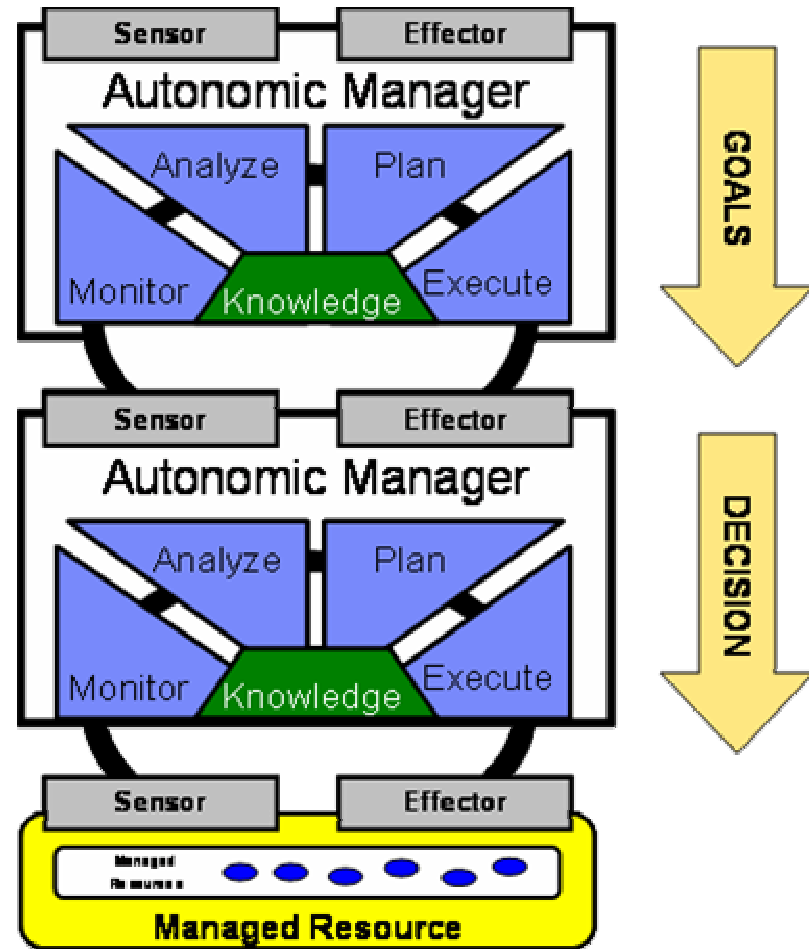
Managed Resource

- Nimmt Entscheidungen entgegen
- Führt entsprechende Handlungen aus

Metapher



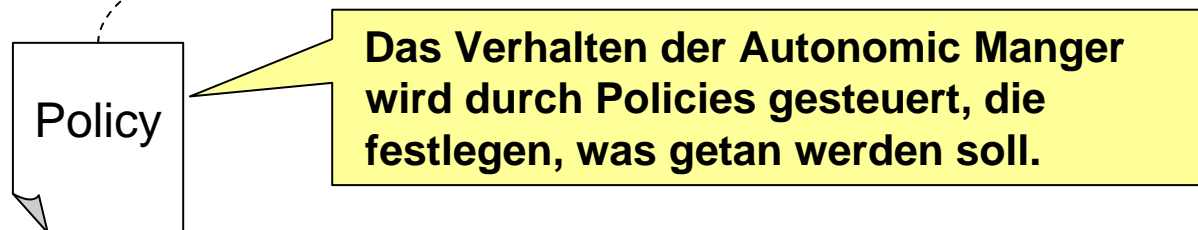
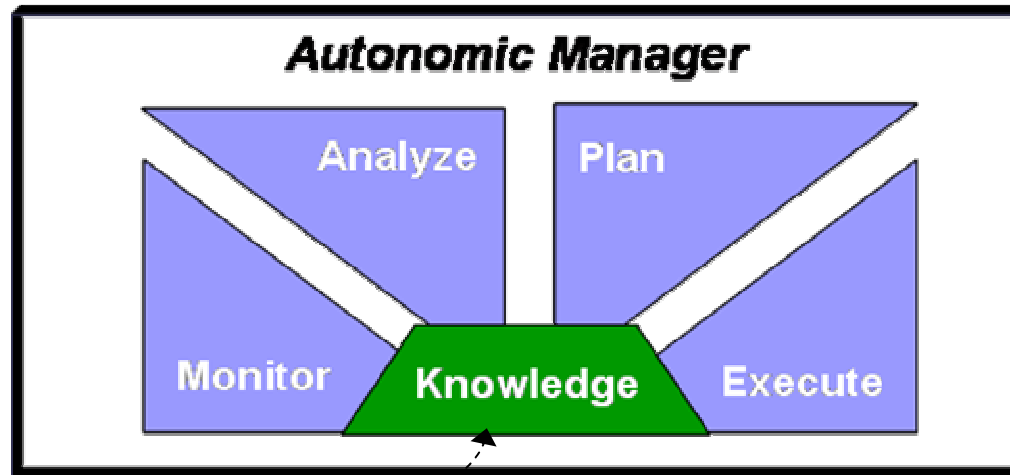
[jkimball-03]



[ibm-04]



Knowledge Service (K-Service)



[ibm-04]



Besondere Herausforderungen:

- Legacy Systeme
- „dumme“ Komponenten

Verwendbare Technologien und Standards:

- Web Services
- Java Management Extensions (JSR3)
- Logging API Specification (JSR47)
- Java Agent Services (JSR87)
- Portlet Specification (JSR168)
- ...

- Motivation
- Grundlagen
- Architektur
- **Produkte**
- Fazit/Ausblick
- Literatur

- IBM: Autonomic Computing Toolkit
 - IBM: Tivoli Produkte
 - IBM: WebSphere Produkte
 - IBM: DB2
 - HP: Utility Data Center Tool
 - SUN Microsystems: “N1” Vision
 - Computer Associates: Sonar
- **Alle Produkte maximal Level 3 (Predictive) bis 4 (Adaptive)**



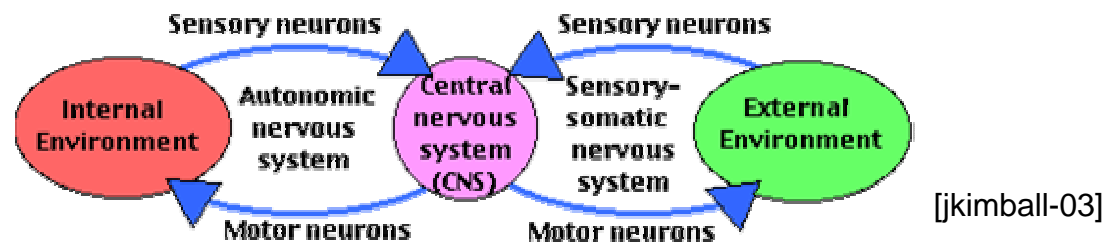
Gliederung

- Motivation
- Grundlagen
- Architektur
- Produkte
- **Fazit/Ausblick**
- Literatur



Zusammenfassung

- Autonomic Computing ist eine Vision, kein Produkt!



- Autonomic Computing ist ein offenes Forschungsgebiet
- Bis zu echtem Autonomic Computing ist es noch ein weiter Weg
 - Evolution, keine Revolution!
- Autonomic Computing ist nur mit offenen Standards realisierbar (bzw. sinnvoll)

Neu?

Nein!

- Fehlererkennung und Fehlertoleranz sind keine neuen Themen
- Besonders aus dem Server und DB Bereich bekannt:
 - Recovery Mechanismen, Load-Balancer, etc.

und JA!

- Globale Sicht auf Unternehmen
- Zusammenspiel vieler Komponenten (keine Insellösungen)

- Schwer umzusetzen
- Evtl. nicht vollständig möglich (→ Level 5 „Autonomic“)
- Hohes Potential
- Positive Entwicklung in den letzten Jahren
 - Immer mehr beteiligte Unternehmen (Autonomic-Computing-Initiative)
 - IEEE Konferenz
 - Unterstützende Standards (z.B. Web Services)
- Evtl. Verlust von Know-How



AW2 Projekt (Ausblick)

- Passt thematisch in jedes Projekt
- Schwerpunkt: Self-Configuration / Self-Healing
- Möglichkeiten im "Pervasive Gaming Framework" Projekt:
 - Architektur: Standardisierte Schnittstellen für AC (z.B. mit Web Services)
 - Aufbau von Knowledge Services und Autonomic Managers
 - Server: Ausfälle oder Fehler kompensieren
 - Clients: Ausfälle oder Fehler kompensieren
 - Clients: Automatische Updates der versch. Softwarekomponenten (Change Management)

- Motivation
- Grundlagen
- Architektur
- Produkte
- Fazit/Ausblick
- **Literatur**



Literatur (Auszug)

- An architectural blueprint for autonomic computing (IBM 2005)
- The dawning of the autonomic computing era (IBM Systems Journal 2003; A. G. Ganek, T. A. Corbi)
- A Vision of Autonomic Computing (IBM 2002)
- Autonomic Computing – Panacea or Poppycock? (Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05); Roy Sterritt, Mike Hinchey)
- Defining Autonomic Computing: A Software Engineering Perspective (Proceedings of the 2005 Australian Software Engineering Conference (ASWEC'05); Paul Lin, Alexander MacArthur, John Leaney)
- Autonomic Computing in Action (IBM 2004; Peter Brittenham)



Weiterführende Links

- <http://www.autonomic-conference.org/>
- <http://www.research.ibm.com/autonomic/overview/>
- <http://www-03.ibm.com/autonomic/>
- <http://www.alphaworks.ibm.com/autonomic?open&ca=dgr-awwikip>
- http://en.wikipedia.org/wiki/Autonomic_Computing
- <http://uebb.cs.tu-berlin.de/lehre/2006SSautonomic-computing/>



Bildnachweis

- [jkimball-03]
<http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/P/PNS.html>
- [ibm-05] An architectural blueprint for autonomic computing (IBM 2005)
- [ibm-04] Autonomic Computing in Action (IBM 2004; Peter Brittenham)
- [ibmred-05] Problem Determination Using Self-Managing Autonomic Technology (IBM Redbook 2005; Manoel et al)

Vielen Dank!

Fragen?