



Workshop über  
Selbstorganisierende, adaptive, kontextsensitive  
verteilte Systeme  
(SAKS 2010)

Assessment –  
Ein Ansatz zur Evaluierung selbstorganisierender Systeme

Jens Tiemann, Mikhail Smirnov und Tanja Zseby

11 Pages

## Assessment – Ein Ansatz zur Evaluierung selbstorganisierender Systeme

Jens Tiemann, Mikhail Smirnov und Tanja Zseby

Fraunhofer Institut für Offene Kommunikationssysteme (FOKUS), Berlin

**Abstract:** Selbstorganisierende Systeme werden in der Kommunikationstechnik als Möglichkeit gesehen, die Dienstqualität zu verbessern und gleichzeitig den Administrationsaufwand zu senken. Die eingesetzten selbstanpassenden Algorithmen führen zu völlig neuen Systemeigenschaften, die beim Testen und der Evaluation der Leistungsfähigkeit dieser Systeme berücksichtigt werden müssen. Ausgehend von Definitionen zu kontextsensitiven Systemen und Überlegungen zu Schnittstellen zwischen der Testumgebung und dem zu testenden System wird das neuartige, als Assessment bezeichnete Evaluationsverfahren vorgestellt. Abschließend werden Probleme bei der Generierung von Testfällen diskutiert und dabei neue, anstehende Forschungsaufgaben skizziert. Der Ansatz soll zur Diskussion einer Systemtheorie dieser neuartigen Systeme beitragen und ganz praktisch das Vertrauen in die Arbeit derartiger, hochgradig flexibler Systeme stärken sowie deren zukünftigen Einsatz fördern.

**Keywords:** Selbstorganisation, Selbstmanagement, Selbstkonfiguration, Selbstoptimierung, Future Internet, Kommunikationssysteme, Testen, Evaluation

### 1 Einleitung

Konzepte der Selbstorganisation und des Selbstmanagement werden für die Anwendung in Computersystemen und zur Steuerung von Kommunikations-Infrastrukturen seit Jahren erforscht. Ihr Einsatz wird in den nächsten Jahren erwartet. Wesentliche Gründe für den Einsatz dieser Konzepte ist die hohe Komplexität zur Kontrolle dieser technischen Systeme. Auch besteht die Erwartung, insgesamt den administrativen Aufwand zu reduzieren (und damit die Kosten zu senken), trotz des weiterhin starken Leistungszuwachses von Computer- und Kommunikationssystemen. Es entstehen aber auch neuartige, komplexe Anwendungen, z.B. im Bereich Opportunistic Radio, die ohne den Einsatz derartiger Mechanismen nicht realisiert werden können.

Im Bereich der Kommunikationstechnik sind seit langem adaptive Algorithmen bekannt, die allerdings vorprogrammiert jeweils nur in wenigen Freiheitsgraden operieren. Das autonome Netzwerkmanagement verlangt jedoch die Einbeziehung einer Vielzahl von Informationen – von der Übertragungsqualität über die Verfügbarkeit von Ressourcen bis hin zu den Anforderungen von Nutzern – sowie eine möglichst umfangreiche Konfiguration der Netzkomponenten. Die Aufgaben des Selbstmanagement in Netzen lassen sich anhand des FCAPS-Modells (Fault, Configuration, Accounting, Performance, Security) beschreiben, das sich mit den bekannten Self-CHOP-Prinzipien (Configuration, Healing, Optimization, Protection) des Autonomic Computing vergleichen lässt.

In diesem Beitrag wird ein Ansatz zur Evaluierung selbstorganisierender, selbstanpassender Kommunikationssysteme (z.B. Mobiltelefone, Basisstationen, Router) vorgestellt. Damit

verbundene Schwierigkeiten und Lösungsansätze werden diskutiert – aufgrund von Systemeigenschaften stellt schon die Kopplung von zu testendem System und Test-System sowie die Initialisierung von Tests eine Schwierigkeit dar. Die vorgeschlagene Evaluierung soll dabei auf bisherigen Tests aufbauen und neue Systemeigenschaften adressieren. Die als Assessment bezeichnete Methode wird kurz in einer Darstellung des Verfahrens skizziert, ein Beispiel soll die Methode illustrieren. Der Beitrag schließt mit einem Vorschlag, wie die Implementierung von einfachen Beispielsystemen forciert werden kann. Dabei entsteht eine Umgebung, die eine Reihe von realistischen Situationen und Aufgaben zur Selbstorganisation für Netzkomponenten präsentiert.

## 2 Selbstorganisierende, kontextsensitive Kommunikationssysteme

Obwohl noch viele Fragen zur konkreten Realisierung selbstorganisierender Systeme offen sind, zeichnen sich doch Gemeinsamkeiten beim Systementwurf ab [MMJ09][ZKH08]:

Die Systeme basieren

- auf einem kognitiven Zyklus (vergleichbar mit dem Sense-Think-Act Cycle als Ansatz der klassischen künstlichen Intelligenz),
- auf der Verarbeitung einer Vielzahl von Umgebungsinformationen oder Kontext,
- auf intern gesammeltem oder vorhandenem Wissen sowie
- auf der Kooperation zwischen Systemen.

Ein früher Einsatz der Prinzipien Selbstorganisation und Selbstmanagement wird im Bereich des Netzmanagements für den Mobilfunk erwartet [SOC][E3]. Gründe hierfür sind die Komplexität der Management-Aufgaben sowie die Möglichkeit, Selbstorganisation in verschiedenen Gebieten und unter verschiedenen Einsatzbedingungen einzuführen. Optimierungsaufgaben können zunächst offline auf vorhandenen Daten des konventionellen Netzmanagements durchgeführt werden und erlauben danach (bei positiven Erfahrungen mit den Algorithmen) automatische Verfahren, die auch in Echtzeit laufen können.

### 2.1. Verwendete Definitionen zu Systemen und Testverfahren

Aufgrund des Ziels dieser Arbeit, selbstorganisierende Systeme zu evaluieren, werden diese Systeme anhand ihres Umgangs mit Kontext definiert [STC09]. Selbstanpassende Systeme (self-adaptive systems) erkennen für den Betrieb notwendigen Kontext und dessen relevante Veränderungen rechtzeitig und passen ihr Verhalten innerhalb einer begrenzten Zeitspanne an. Hierunter fallen auch Systeme, die einen sehr großen Satz von festen Regeln enthalten. Kognitive Systeme (cognitive systems) sind selbstanpassende Systeme, die zusätzlich in der Lage sind, neuen relevanten Kontext zu entdecken und neue Verhaltensweisen zu entwickeln, z.B. durch Lernen. Darauf aufbauend sind ich-bewusste Systeme (self-aware systems) definiert, die sowohl die Relevanz von Kontext, seine zeitige Verarbeitung als auch die daraus abgeleiteten Verhaltensweisen evaluieren können. Derartige Systeme spielen für diese Arbeit zur Evaluation noch keine Rolle.

Kommunikationssysteme durchlaufen verschiedene Tests, wie z.B. Konformitäts- und Leistungsmessungen, die natürlich auch für selbstorganisierende Systeme relevant sind. Dabei ist das Ergebnis eines Konformitätstests abhängig von der *Spezifikation* und der *Implementierung*, während eine Leistungsmessung zusätzlich von den aktuellen Betriebsbedingungen abhängt, im Folgenden auch *Situation* genannt. Beispielsweise kann der

Durchsatz eines Kommunikationssystems von den auftretenden Paketlängen abhängen. Als Erweiterung ist damit das Ergebnis des Assessments sowohl von *Spezifikation*, *Implementierung* und der *Situation* abhängig, zusätzlich aber auch vom komplexen, *internen Zustand* des zu testenden Systems, z.B. dem erlernten Wissen.

## 2.2. Bestehende Ansätze zum Testen selbstorganisierender Systeme

Erste Ansätze zur Evaluation stammen aus dem Bereich Autonomic Computing. Verglichen mit den Tests im Bereich Autonomic Communication ergeben sich Gemeinsamkeiten beim Ziel, die Leistung von Systemen zu bestimmen. Es existieren einige Ansätze zur Beurteilung von autonomen Systemen, die sich nur auf eine Auswertung von Leistungsmessungen beschränken. Im besten Fall werden diese Metriken gruppiert, trotzdem ist meist nicht klar, ob die vorgestellten Metriken vollständig und systematisch erhoben sind. Auch werden interne und externe Metriken sowie Metriken verschiedener Abstraktionsniveaus vermischt, was zu Problemen bei der Evaluation von realen Systemen führen wird (z.B. fehlender Zugriff auf interne Ressourcen).

Brown et al. [BHH04] beschreiben einen ersten Ansatz für den Einsatz von Benchmark-Verfahren bei autonomen Eigenschaften. Das Ziel ist die Erstellung einer Benchmark-Suite für die Self-CHOP-Eigenschaften, um sowohl die Einführung derartiger Systeme zu fördern als auch den Vergleich zwischen Systemen zu ermöglichen. Die vorgeschlagene Methode basiert auf einer Test-Umgebung zur Generierung von synthetischer Last und auf dem Einbringen von zusätzlichen Änderungen (z.B. Störungen oder Änderungen an Ressourcen). Das Ergebnis dieser Evaluierung besteht aus der Messung von Systemantworten (z.B. Reaktionszeit, Qualität, Kosten) und wird in einem Kennzahlensystem präsentiert. Als Herausforderungen wird das Einfügen der Änderungen identifiziert, z.B. deren Wiederholbarkeit oder ihre Repräsentativität.

Das Forschungsprojekt SOCRATES entwickelte Metriken, die zur Beurteilung selbstorganisierender Mobilfunk-Netze geeignet sind [ALS08]: Leistungsaspekte, Abdeckung, Kapazität, betriebswirtschaftliche Aspekte wie Investitionsaufwand (CAPEX) und Betriebsaufwand (OPEX), sowie zusätzliche, von den Algorithmen abhängige Metriken wie Konvergenzzeit oder Robustheit. Ziel ist die Bewertung des Zugewinns durch die Nutzung von neuen, selbstanpassenden Algorithmen. Die Metriken werden anhand der Systemaspekte gruppiert und im Sinne einer Leistung des Systems mithilfe von Benchmarks ermittelt. Das Gesamtergebnis wird aus einzelnen Gruppen der erwähnten Metriken und deren Gewichtung ermittelt. Die Gewichtung kann dabei von dem Geschäftsmodell beim Einsatz der Systeme abhängen, z.B. ob eher der Preis oder die Qualität von Diensten im Vordergrund steht.

## 3 Assessment

Mit dem hier vorgestellten Ansatz möchten wir auch die Adaptierung zwischen Tester und zu testendem System für Kommunikationssysteme behandeln und die Frage der Wiederholbarkeit von Tests adressieren. Dabei soll es nicht um die Messung einzelner oder Gruppen von Leistungsindikatoren gehen. Vielmehr werden wir aus einer externen Sichtweise ein „Level of Autonomy“ [LSK06] ermitteln, aufgrund dessen die Autonomie des Systems und damit die Möglichkeiten und Grenzen des Selbstmanagements von Kommunikationssystemen vor deren Einsatz abgeschätzt werden kann. Dabei wird zu jeder Zeit berücksichtigt, dass das System korrekt angepasst ist (Process Correctness [DDF06]), d.h. jeder Anpassungsschritt muss die

korrekte Funktion des Systems erlauben. Von vielen, kommerziell eingesetzten Systemen und Algorithmen liegt die innere Struktur nicht offen, deshalb wird der Ansatz des Black-Box-Testens gewählt. Ein weiterer Vorteil unseres Testansatzes ist auch, dass die Art des Algorithmus keine Rolle spielt und selbstorganisierende Systeme mit konventionellen Systemen auch auf dieser Funktionsebene miteinander verglichen werden können. Es wird nicht die Lernfähigkeit evaluiert, sondern ausschließlich die Fähigkeit, anwendungsbezogene Probleme zu lösen.

Der Vergleich mit der Methode *Assessment Center* bei der Auswahl von Bewerbern bietet sich an: Als möglicher Teil dieser Methode werden z.B. einem Kandidaten verschiedene, zu lösende Aufgaben präsentiert, und es kann sowohl sein Vorgehen als auch das Ergebnis bewertet werden. In dieser Art von Planspielen lassen sich mit vergleichsweise wenigen Parametern bereits komplexe und repräsentative Situationen erzeugen – insbesondere bei den Kandidaten wird eine realistische Problemlösungs-Situation erzeugt.

Für die praktische Evaluation von selbstanpassenden Systemen, dem Assessment-Prozess, werden Methoden aus der Softwaretechnik und Methoden zum Testen von Kommunikationssystemen kombiniert. Wie im Software-Entwurf wird von Anwendungsfällen (UC - Use Cases) ausgegangen, aus denen Testfälle (TC - Test Cases) abgeleitet werden. Diese Testfälle bilden ein Benchmark-Verfahren, wie man sie ebenfalls aus der Softwaretechnik kennt. Vom Testen von Kommunikationssystemen übernehmen wir die Methoden der Generierung von Testdatenverkehr und die Beobachtung des Systemverhaltens.

### 3.1. Assessment-Schnittstellen

Eines der Hauptprobleme ist die Vielzahl an Informationen, die den Entscheidungsprozess eines selbstorganisierenden, kontextsensitiven Systems beeinflussen. Die Systemantwort eines lernenden Systems ist nicht mehr in jedem Fall einfach vorherzusagen und hängt auch von internen Algorithmen und bereits gesammelten Informationen ab. Hilfreich ist hier auch für das Testen und die Evaluierung der Einsatz eines Systems zum Kontextmanagement (Context Management System), über dessen Schnittstellen die Testumgebung und das zu testende System (SuT – System under Test) miteinander verbunden werden können.

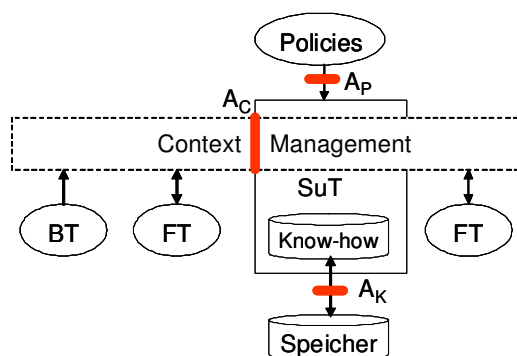


Abbildung 1: Schnittstellen für das Assessment

Dabei kann das SuT für einen einzelnen Algorithmus, für ein Teilsystem oder ein komplettes Gerät stehen. In einem einfachen Modell wird das System durch Policies gesteuert

(beispielsweise werden definierte Frequenzbereiche zur Nutzung freigegeben) und der Einfluss auf das System kann über die Schnittstelle  $A_P$  kontrolliert werden. Das System reagiert auf Umgebungsbedingungen und Nutzungsanforderungen in Form von Kontext, hier dargestellt durch die Schnittstelle  $A_C$ , und baut dabei intern einen komplexen Systemzustand auf (beispielsweise das Wissen, das ein lernender Algorithmus ansammelt). Da Algorithmen und andere Systeminterna unbekannt sein können, geht das Assessment von dem Black-Box-Testansatz aus. Das Systemverhalten ist hochgradig von diesem internen Systemzustand abhängig, daher ist eine Schnittstelle nötig, mit der das zu testende System in definierte Zustände gebracht werden kann ( $A_K$ ). Hierüber kann das System initialisiert werden, und es wird überhaupt erst möglich, einen Test effizient zu wiederholen.

Im Fall von kontextsensitiven Systemen muss die Situation oder die Umgebungsinformation nicht direkt an den Sensoren erzeugt werden, sondern es können über das Kontextmanagement auch künstliche Kontextquellen als Testgeneratoren eingebunden werden. Hier wird, wie beim Testen von Kommunikationssystemen, zwischen Vordergrundtestern (FT) und Hintergrundtestern (BT) unterschieden. Das zu testende System interagiert mit den komplexeren Vordergrundtestern, d.h. es beeinflusst durch seine Reaktion wiederum den Tester, während die einfacher aufgebauten Hintergrundtester sich hiervon nicht beeinflussen lassen und eingesetzt werden, um eine Grundlast zu erzeugen.

Bei kontextsensitiven Systemen kann auf die tatsächliche Generierung von Datenströmen verzichtet werden, wenn sich mittels des Kontextmanagements die zu untersuchenden Algorithmen eines Systems von Sensoren und anderen Kontextquellen isolieren lassen. Auf diese Weise lassen sich wesentlich einfacher sehr komplexe Situationen generieren, da man mittels des Kontextmanagements quasi interne Schnittstellen eines Systems zum Testen nutzen kann. Beispielsweise könnte ein interner Monitor zur Analyse des Datenverkehrs durch einen Test-Generator ersetzt werden, der (Test-)Verkehrsströme als schon ausgewertete Information anbietet. Da auch die Generierung von Testdatenströmen auf dieser abstrakten Beschreibung von Verkehrsströmen basiert, lässt sich eine aufwändige Erzeugung von tatsächlichem Datenverkehr vermeiden.

### 3.2. Assessment-Prozess

Beim Assessment-Prozess [E3D23] wird von den Anwendungsfällen (UC) ausgegangen, die auch der Implementierung des zu testenden Systems zugrunde liegen.

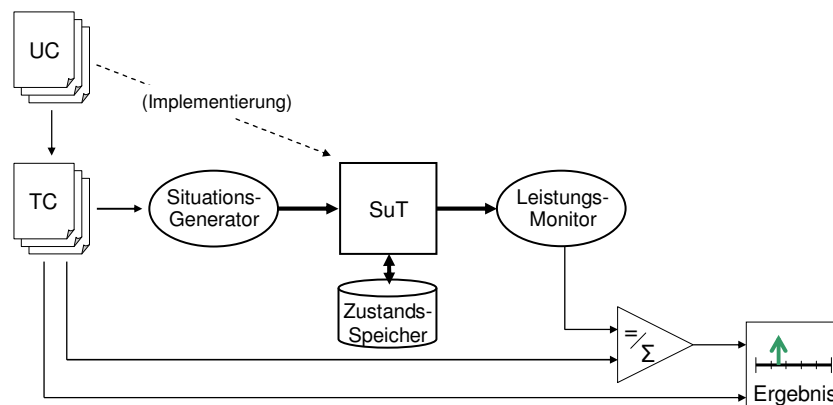


Abbildung 2: Assessment-Prozess

Zur Erstellung der Testfälle (TC) werden den Anwendungsfällen Angaben zum Kontext und dem erwarteten Systemverhalten entnommen. Im Idealfall sind diese direkt verknüpft mit einer Metrik und Grenzwerten zur erwarteten Leistungsfähigkeit des Systems. Die Testfälle stellen Situationen oder zu lösende Probleme dar. Über die Schnittstelle  $A_K$  wird das System für die jeweiligen Tests aus einem externen Speicher für Systemzustände vor einem neuen Testfall ggf. initialisiert. Die Testfälle selber werden von Kontext- oder Situationsgeneratoren erzeugt und mehrfach durchlaufen, da sich ein selbstorganisierendes System mithilfe eines selbstanpassenden Algorithmus im Laufe des Betrieb verändern kann, d.h. es wird die Anpassung des SuT an eine Situation ermöglicht. Das Verhalten des Systems wird über normale Messungen der Leistungsfähigkeit aufgenommen (z.B. der Durchsatz für einen bestimmten Dienst in einer definierten Qualität). Die Leistung des Systems kann dabei über einfache Metriken bestimmt werden oder aus schon vorverarbeiteten Werten bestehen, wie sie Key Performance Indicator (KPI) bieten. Die Ergebnisse für einen Testfall werden gesammelt, und eine Aufgabe gilt als gelöst, wenn mindestens 50% der Läufe erfolgreich waren. Ein solcher Testfall kann auch als Benchmark gesehen werden, der eine Aufgabe einer definierten Schwierigkeit enthält. Das System wird abschließend mit dem höchsten Schwierigkeitsgrad einer Reihe von Benchmarks bewertet, die das System lösen konnte.

### 3.3. Assessment-Beispiel

Mit diesem Beispiel soll der Assessment-Prozess illustriert werden. Dabei kommen sowohl Demo-Algorithmen als auch Demo-Testfälle zum Einsatz, deren Konstruktion den Assessment-Vorgang deutlich sichtbar macht. Eine Implementierung des Assessment-Verfahrens für reale Systeme befindet sich in der Entwicklung.

In diesem Beispiel soll ein Teilsystem untersucht und evaluiert werden, das die selbständige Kanal-Zuweisung für eine Basisstation oder einen WLAN-Zugangspunkt ermöglicht. Dabei kommt es darauf an, dass der gleiche Kanal nicht zweimal in benachbarten Zellen des typischen „7-Zellen“-Musters genutzt wird. In diesem stark vereinfachten Beispiel wird also eine Umgebung oder Situation erzeugt, in der 6 Zellen jeweils einen Kanal nutzen. Das zu testende System soll nun mit seinem implementierten Algorithmus einen weiteren Kanal im Zentrum auswählen.

In diesem Beispiel werden drei Demo-Fälle mit steigender Schwierigkeit verwendet:

- „Umfangreiche Ressourcen“ – es sind 50 Kanäle verfügbar, damit wird auch eine zufällige Auswahl eines Kanals durch die SuT in vielen Fällen erfolgreich sein.
- „Knappe Ressourcen“ – es sind genau 7 Kanäle verfügbar, der nicht belegte Kanal muss also von dem SuT per Messung bestimmt und ausgewählt werden.
- „Störer“ – es sind 9 Kanäle verfügbar, allerdings springt eine Zelle bei der Kanalnutzung zwischen Kanal 6, 7 und 8. Damit muss das Verhalten des Störers in der Nachbarschaft von dem SuT „gelernt“ werden, um erfolgreich einen Kanal zu wählen.

Die drei verwendeten Demo-Implementierungen des Algorithmus zur Kanalauswahl sind damit:

- „Random“ – zufällige Auswahl eines Kanals;
- „Sense“ – die Kanalnutzung in der Umgebung wird einmal zu Beginn gemessen und ein ungenutzter Kanal wird ausgewählt;



- „Learn“ – die Wahrscheinlichkeit der Kanalnutzung der Nachbarn wird durch fortlaufende Messungen bestimmt, der Kanal mit der geringsten Nutzungswahrscheinlichkeit wird ausgewählt.

Nun werden die drei Algorithmen den drei Demo-Situationen ausgesetzt. Bei jeweils mehreren Läufen wird eine deutliche Differenzierung in Bezug auf die Problemlösefähigkeit der Algorithmen erwartet. Der Algorithmus „Random“ kann nur als Funktionstest gesehen werden, bei umfangreich verfügbaren Ressourcen genügt er allerdings einfachen Ansprüchen. Wird tatsächlich durch das SuT die Kanalauswahl in der Nachbarschaft gemessen, so ermöglicht der Algorithmus „Sense“ eine Kanalauswahl in einer statischen Umgebung. Erst der Algorithmus „Learn“ erlaubt auch eine Anpassung an dynamische und damit komplexere Umgebungen und Situationen.

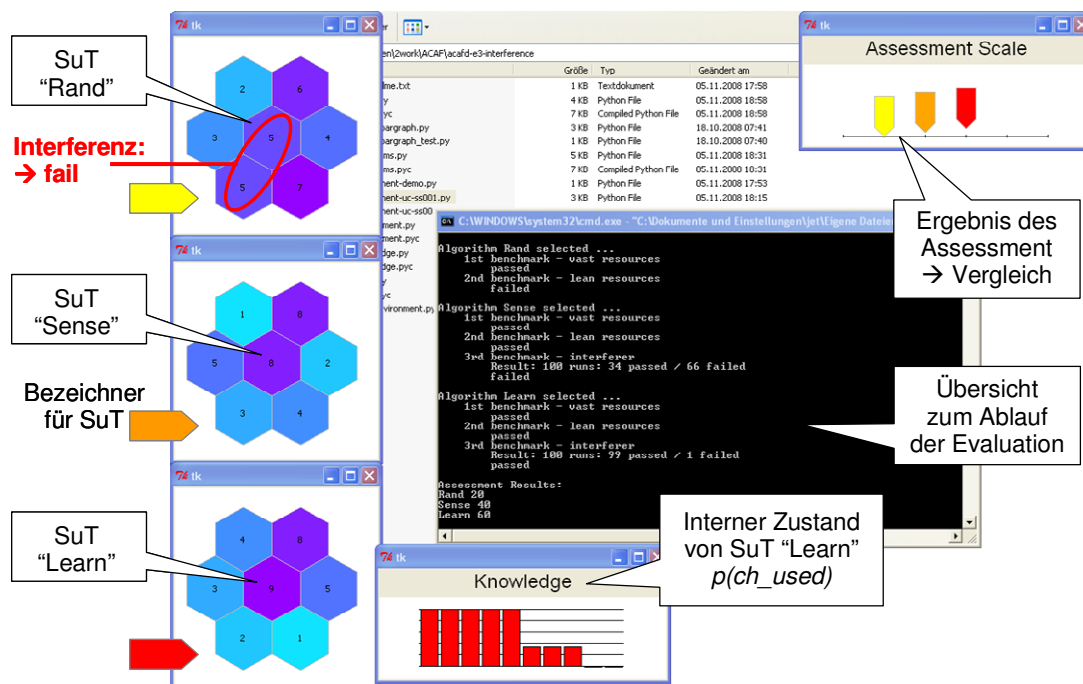


Abbildung 3: Assessment-Beispiel

Die Abbildung zeigt die Durchführung und das Ergebnis des Assessments für die drei Demo-Implementierungen unter Einsatz der drei Demo-Testfälle. Auf der linken Seite werden die Umgebungen der SuT dargestellt, die jeweils die Kanalauswahl in der Mitte des Zellenmusters vornehmen müssen. Wählt der Algorithmus einen schon vorhandenen Kanal aus (Interferenz), so gilt dieser Test-Lauf als nicht bestanden. Im Fall des „lernenden“ Algorithmus wird auch der interne Zustand des SuT dargestellt. Im abgebildeten Beispiel wurden jeweils 100 Läufe pro Testfall durchgeführt, bei mehr als 50 erfolgreichen Läufen gilt ein einzelner Testfall als bestanden und der nächste, schwierigere wird ausgeführt. Das Ergebnis ist hier eine vergleichende Darstellung der evaluierten Algorithmen (oben rechts im Bild).



## 4 Generierung komplexer Situationen

Ein wesentlicher Bestandteil des Assessment-Verfahrens ist die Erstellung von Testfällen oder Benchmarks, die verschiedene und definierte Schwierigkeitsgrade bei der Problemlösung abdecken müssen. Neben der Problembeschreibung spielt auch die erwartete Leistung des Systems eine Rolle: Von dem besseren System wird schneller eine Lösung des Problems angeboten. Im einfachsten Fall stellen Experten diese Benchmarks zusammen und orientieren sich dabei an bisherigen, konventionellen Einsatzgebieten und der Leistungsfähigkeit verfügbarer Systeme. Allerdings kann die Ordnung von Aufgaben im Bereich mittlerer Schwierigkeiten Probleme bereiten. Bei diesen Benchmarks könnte die Schwierigkeit auf andere Größen zurückgeführt werden, wie z.B. die Komplexität (einer Aufgabe oder Situation). Sehr schwierige Benchmarks lassen sich an theoretischen Maximalwerten orientieren (z.B. die maximal mögliche Auslastung von Ressourcen), während sehr leichte Benchmarks eher die Rolle von Funktionstests übernehmen (im Sinne der Process Correctness).

Eine andere Möglichkeit zur Ordnung von Benchmarks ist eine Reihe von Implementierungen, die über die Ausführung verschiedener Benchmarks miteinander verglichen werden können. Ähnlich dem Intelligenztest für Menschen kann bei hinreichend vielen verfügbaren und vergleichbaren Implementierungen (eingeschränkt auf einen speziellen Anwendungsbereich) eine statistische Aussage zu einzelnen Aspekten der Fähigkeiten eines Systems gemacht werden, basierend auf latenten, nicht direkt messbaren Variablen. Hierbei ergibt sich eine relative Aussage zur Schwierigkeit von Benchmarks und den Fähigkeiten von Algorithmen und Systemen. Dies ist für einen Vergleich von Systemen im Sinne der erwarteten Leistungsfähigkeit ausreichend.

Um sowohl implementierte Systeme zur Verfügung zu haben als auch realistische, komplexe Situationen analysieren zu können, eignet sich auch ein Wettbewerb ähnlich dem RoboCup (Turnier für Roboter-Fußball). Dabei wird davon ausgegangen, dass schon eine geringe Anzahl von Regeln, Umgebungsvariablen und implementierter Funktionalität ausreicht, um komplexe Situationen zu erzeugen, insbesondere in der Interaktion verschiedener Systeme untereinander und mit ihrer Umgebung. Im Bereich der Kommunikationstechnik bieten sich dabei Aufgaben an, bei denen qualitativ hochwertige Kommunikationsdienste auf einem Netz mit beschränkten Ressourcen angeboten werden: Dabei müssen die verschiedenen Systeme um die gleichen Ressourcen konkurrieren und damit auch interagieren. Verschiedene Schwierigkeitsgrade lassen sich über die Netztopologie (z.B. Anordnung und Mobilität von Netzknoten, heterogene Verbindungen) und den Dienste-Mix erreichen. Ein solcher Wettbewerb kann als Simulation oder mit vereinfachten realen Systemen durchgeführt werden. Ein Schlüssel ist hier wieder die Verwendung eines gemeinsamen Systems zum Kontextmanagement.

Trotz der vereinfachten Bedingungen wird erwartet, dass das Assessment-Verfahren auf diese Weise überprüft und verfeinert werden kann. Auch trägt ein solcher Wettbewerb insgesamt zum besseren Verständnis von selbstorganisierenden Kommunikationssystemen bei.

## 5 Zusammenfassung und Ausblick

Zum vollständigen Verständnis von komplexen Systemen und deren Entwurf gehören das Testen und die Bestimmung der Leistungsfähigkeit. Selbstorganisierende, selbstanpassende Systeme erfordern allerdings eine zusätzliche Evaluation im Zusammenhang mit ihrer Anpassungsfähigkeit: *Inwieweit ist ein System in der Lage, spezifische Probleme innerhalb seines Einsatzgebietes zu lösen?* Da der Betrieb dieser Systeme auf einer Vielzahl von Informationen (Kontext) und dem angesammelten, internen Wissen beruht, sind zum Testen und zur Evaluierung besondere Mechanismen nötig. Die vollständige Evaluation eines selbstorganisierenden Kommunikationssystems muss über bisher verwendete Ansätze, wie das Testen der Konformität oder die Messung der Leistungsfähigkeit, hinausgehen. Insbesondere muss es eine neue Systemeigenschaft erfassen – die Anpassung an die Betriebsumgebung und -bedingungen. Diese verborgene Eigenschaft kann nicht direkt gemessen werden, sondern wird aus Leistungsmessungen abgeleitet. Die vorgestellte Methode des Assessments geht dabei von einer eindeutigen, externen Sichtweise auf das zu testende System aus.

Die vorgeschlagene Evaluationsmethode, das Assessment, beruht auf der Durchführung einer Reihe von Benchmark-Tests, die nach der Schwierigkeit des zu lösenden Problems sortiert sind. Durch diese Methode wird die Bestimmung der Anpassungsfähigkeit auf die Fähigkeit zur Problemlösung in verschiedenen komplexen Situationen abgebildet.

Die Komplexität einer Aufgabe zu bestimmen ist allerdings nicht einfach – sie kann zunächst durch Experten eingeschätzt werden und sich auch an theoretischen Grenzen orientieren. Besser wäre eine automatische Ableitung der Benchmarks, z.B. indem beobachtet wird, wie selbstorganisierende Systeme mit anderen Systemen und einer mehr oder weniger komplexen Einsatzumgebung interagieren.

Das Ziel der Arbeiten ist die Entwicklung von beispielhaften, anerkannten Testsuiten zum Vergleich der Leistungsfähigkeit selbstorganisierender Systeme, die dann eine Zertifizierung dieser neuen Systemeigenschaften ermöglicht.

## Danksagung

Diese Arbeiten wurden teilweise im 7. EU-Forschungsrahmenprogramm innerhalb des Projekts E<sup>3</sup> (End-to-End Efficiency) durchgeführt. Dieser Beitrag stellt nur die Sichtweise der Autoren dar, die Gemeinschaft kann für jegliche Verwendung der hierin enthaltenen Informationen nicht haftbar gemacht werden. Die Autoren danken für den kollegialen Austausch innerhalb des E<sup>3</sup>-Konsortiums.

## Literatur

- [ALS08] M. Amirijoo, R. Litjens, K. Spaey, M. Döttling, T. Jansen, N. Scully and U. Türke: "Use cases, requirements and assessment criteria for future self-organising radio access networks", IWSOS 2008, Vienna, Austria, December 10-12, 2008
- [BHH04] A. B. Brown, J. Hellerstein, M. Hogstrom, T. Lau, S. Lightstone, P. Shum and M. Peterson Yost, "Benchmarking autonomic capabilities: promises and pitfalls", First International Conference on Autonomic Computing (ICAC'04), pp. 266-267, 2004
- [DDF06] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt and F. Zambonelli, "A survey of autonomic communications," ACM Trans. Auton. Adapt. Syst. 1, 2 (Dec. 2006), pp. 223-259
- [E3] Projektseite "E3 – End-to-End Efficiency", <https://ict-e3.eu/>
- [E3D23] Project E3 Deliverable D2.3, "Architecture, information model and reference points, assessment framework, platform independent programmable interfaces", September 2009
- [LSK06] D. Lewis, D. O'Sullivan and J. Keeney, "Towards the knowledge-driven benchmarking of autonomic communications," In Proceedings of the 2006 international Symposium on on World of Wireless, Mobile and Multimedia Networks (June 26 - 29, 2006). International Workshop on Wireless Mobile Multimedia. IEEE Computer Society, Washington, DC, USA
- [MMJ09] A. Marikar, J. Mödecker, K. Jonas, "Selbstverwaltung im Future Internet", Electronic Communications of the EASST, Volume 17: Kommunikation in Verteilten Systemen 2009
- [TT09] S. Taranu and J. Tiemann, "General method for testing context aware applications," In Proceedings of the 6th international Workshop on Managing Ubiquitous Communications and Services (Barcelona, Spain, June 15 2009). MUCS '09. ACM, New York, NY, USA
- [SOC] Projektseite "SOCRATES - Self-Optimisation & self-ConfiguRATion in wirelEss networkS", <http://www.fp7-socrates.org/>
- [STC09] M. Smirnov, J. Tiemann, R. Chaparadza, Y. Rebahi, et al.: "Demystifying self-awareness of autonomic systems", ICT-MobileSummit 2009. Conference Proceedings, 10-12 June 2009, Santander, Spain, IIMC, Dublin, Ireland, ISBN: 978-1-905824-12-0 9
- [ZKH08] T. Zseby, M. Kleis, T. Hirsch, „The Internet of the Future: Challenges, Approaches and Obstacles“, Telekommunikation Aktuell, Vol.62 (2008), No. 5-6, pp. 1-19, Erlangen, Germany