Zuverlässige adaptive Informationssysteme

Motivation und Begriffsbildung

Matthias Kerbeck

Seminararbeit

Betreuer: Boris Stumm

Technische Universität Kaiserslautern
Fachbereich Informatik
Kaiserslautern, 13. Januar 2006

Diese Arbeit beschäftigt sich mit den Anforderungen an zuverlässige adaptive Informationssysteme (DAIS). Dabei geben anwendungsbezogene Beispiele einen Eindruck über das Einsatzgebiet aktueller DAIS. Anhand dieser Beispiele werden die extremen Anforderungen diskutiert, die an solche Systeme gestellt werden, sowie die zugrunde liegenden Termini erläutert und von einander abgegrenzt. Anschließend werden Ziele und Konzepte für Entwurf und Implementierung von DAIS vorgestellt und die beiden zentralen Aspekte Zuverlässigkeit und Flexibilität einander gegenübergestellt. Ein Ausblick auf Entwicklungen und Herausforderungen künftiger DAIS rundet diese Arbeit ab.

Inhaltsverzeichnis

1 Einführung

Nach Ansicht von Soziologen entwickelt sich unsere Gesellschaft zu einer „Informationsgesellschaft“ – ein Begriff, der widerspiegelt, wie sehr die Information, vor allem in hoch entwickelten Ländern, zu einem wichtigen Wirtschaftsfaktor geworden ist. Dabei werden Verwaltung sowie adäquates Bereitstellen von Informationen für Wirtschaft und Gesellschaft von immer größerer Bedeutung. Nahezu jede größere Organisation bzw. jedes größere Unternehmen nutzt so genannte *Informationssysteme* zur Informationsverwaltung. Dabei müssen viele in einem Unternehmen anfallende sowie externe Informationen in ein ganzheitliches Informationssystem integriert und dessen Benutzern bereitgestellt werden. Kennzeichnend für heutige Informationssysteme sind komplexe, teils langlebige *Arbeitsdurchläufe* (engl.: workflows), welche betriebliche Prozesse widerspiegeln. Aufgrund ihrer Wichtigkeit für Wirtschaft und Gesellschaft gibt es an diese Informationssysteme besonderen Anforderungen.

2 Informationssysteme kritischer Infrastruktur-Einrichtungen

Ein anwendungsbezogener Überblick über heutige Informationssysteme soll an dieser Stelle das notwendige Hintergrundwissen liefern für die anschließende Diskussion der Anforderungen an solche Systeme. Als Beispiele werden im Folgenden zwei so genannte „kritische Infrastruktur-Einrichtungen“ vorgestellt werden, deren zentrale Komponenten computergestützte Informationssysteme sind. Die Kenntnis der Charakteristika solcher Systeme ist notwendig, um die zugrunde liegende Technologie verstehen, entwickeln und verbessern zu können. Die Beispielsysteme umfassen die Bereiche Finanz- und Schienenverkehrswesen.

2.1 Das Bezahlsystem in den USA

Bezahlsysteme (engl.: payment systems) sind zentrale Komponenten des heutigen Währungs- und Finanzwesens, von denen das effektive Funktionieren der Volkswirtschaften vor allem der hoch entwickelten Länder der Erde abhängt. Ein Bezahlsystem umfasst ein Netzwerk von Banken, Zentralbanken, Firmen und staatlichen Institutionen und ein computergestütztes Netzwerk für das reibungslose Abwickeln von Geschäfts- und Finanztransaktionen, wie beispielsweise Gehalts- und Lohnzahlungen, Cashflows für das Beschaffen von Rohstoffen im produzierenden Gewerbe oder Überweisungen an Renten- und Finanzmärkten.

Schlüsselelemente eines Bezahlsystems sind dabei die Finanzdienstleistungen, die von Kreditinstituten angeboten werden. Die finanziellen Transaktionen sind bargeldlose Geldbewegungen, die über Kreditinstitute abgewickelt werden. Die Zahlungen erfolgen in Form von Buchgeld zwischen Giro- bzw. Kontokorrentkonten. Das Konto des Auftraggebenden wird dabei mit einem entsprechenden Betrag belastet, ein oder mehrere Empfänger erhalten demgemäß eine Gutschrift auf ihren Konten. Die Kreditinstitute erheben für solche Transaktionen zumeist eine Gebühr.

Federal Reserve Bank: Die Zentralbank der USA. Die amerikanische Zentralbank besteht aus zwölf regionalen Banken und ist das wichtigste Element im US-Bezahlsystem. Da es bei einem hoch komplexen Bankwesen wie dem der USA sehr ineffizient wäre, zu Verrechnungs- und Überweisungszwecken eine Vielzahl von bilateralen Beziehungen unter vielen Banken zu führen und viele Konten bei einer großen Anzahl anderer Banken zu haben, besteht eine wichtige Aufgabe der Federal Reserve darin, als zentrale Instanz Dienstleistungen für das Führen von Konten für die übrigen Banken der USA anzubieten. Dies ist ökonomischer als ohne Zentraleinheit, da das Führen von Konten teuer ist, aber jede Bank im Stande sein muss, jederzeit Geld überweisen bzw. empfangen zu können, von jeder anderen Organisation oder Person, die ein Konto bei einer anderen Bank führt. Fast alle größeren kommerziellen Banken der USA sind „Mitglied" mit direkter Anbindung an die Federal Reserve Bank, das sind derzeit mehr als 9500. Kleinere Nicht-Mitglieder erlangen Zugang durch entsprechende Anbindung an eine größere Bank, die Mitglied ist.

Über das Federal-Reserve-Bank-System werden bargeldlose Transaktionen auf elektronischem Wege getätigt, wie beispielsweise bei den im Folgenden beschriebenen Großhandelszahlungen via *Fedwire* oder bei Überweisungen zum Begleichen eines Schecks.

Die Federal Reserve Bank ist über ein Hochgeschwindigkeits-Telekommunikationsnetzwerk namens Fednet mit ihren Mitgliedsbanken verbunden, die mithilfe der proprietären Software der Federal Reserve Bank, mit Namen Fedline Station, auf das System und ihre Konten bei der Federal Reserve Bank zuzugreifen. Wenn nun beispielsweise eine Gesellschaft oder eine Person eine Überweisung eines bestimmten Geldbetrags tätigen wollte, würde sie ihre Bank dementsprechend anweisen; Diese würde dann mithilfe der Fedline Station die Transaktion tätigen.

Das US-Bezahlsystem besteht aus zwei größeren Gruppen von Systemen, die im Folgenden näher betrachtet werden, nämlich aus solchen für Großhandels- sowie für Einzelhandelszahlungen.

Großhandels-Bezahlsysteme. Bei Großhandelszahlungen werden im Schnitt etwa 3 Millionen Dollar überwiesen. Sie werden von Kreditinstituten und großen Unternehmen getätigt. Das in erster Linie für inländische Transaktionen genutzten System ist Fedwire.

Fedwire wird von der Federal Reserve Bank betrieben und ermöglicht den angeschlossenen Kreditinstituten die Überweisung von Geldbeträgen von ihren eigenen Konten bei der Zentralbank oder von Konten ihrer Kunden. Auch das amerikanische Finanzministerium und diverse Bundesbehörden nutzen Fedwire, um finanzielle Transaktionen zu tätigen. Die Finanztransaktionen werden dabei mithilfe des Fednet vollzogen.

Der folgende Ablauf illustriert beispielshaft eine Finanztransaktion via Fedwire [KEF+98]:

- Kunde A möchte eine Million Dollar von seinem Konto bei Bank A auf Kunde B's Konto bei Bank B überweisen.
- Bank A überweist daraufhin eine Million Dollar, indem sie eine Nachricht via Fedwire an die Federal Reserve Bank versendet, mit der sie diese anweist und autorisiert, das Konto von Bank A bei der Federal Reserve Bank mit einer Million Dollar zu belasten und diese Summe dem Konto von Bank B bei der Federal Reserve Bank gutzuschreiben.
- Ist die genannte Summe dem Konto von Bank B gutgeschrieben, ist der Kapitaltransfer via Fedwire beendet; Für direkt an Fedwire angeschlossene Banken bewegt sich die Dauer einer solchen Transaktion im Bereich von Sekunden.
- Wenn die Fedwire-Transaktion abgeschlossen ist, schreibt Bank B die überwiesene Summe ihrem Kunden B auf dessen Konto noch am selben Tag gut.

Tabelle 1 gibt einen Überblick über die Zahl der Nachrichten und Höhe der Summen, die täglich via Fedwire versandt werden, anhand dem man die immensen Werte ermessen kann, die jedes Jahr über Fedwire versandt werden.

Tabelle 1. Fedwire-Statistik von 2001 bis 2004 [FedR04].

	2001	2002	2003	2004
Durchschnittliche Anzahl Nachrichten pro Tag	448.030	458.084	491.158	494.479
Durchschnittlich transferierte Summe pro Tag	1,69 Billionen Dollar	1,62 Billionen Dollar	1,74 Billionen Dollar	1,86 Billionen Dollar
Durchschnittliche Summe pro Transfer	3,77 Millionen Dollar	3,53 Millionen Dollar	3,54 Millionen Dollar	3,76 Millionen Dollar

Einzelhandelszahlungen. Bei einer Einzelhandelszahlung werden zumeist Kleinbeträge überwiesen; Dies sind im Schnitt 25 Dollar. Sie werden von Einzelpersonen oder Unternehmen beim Erwerb von Waren oder der Inanspruchnahme von Dienstleistungen getätigt. Beispiele für Einzelhandels-Bezahlsysteme sind die Verwendung von Schecks und Kreditkarten; Der Ablauf bei der Verwendung letzterer ist im Folgenden beispielhaft dargelegt.

Folgende Schritte umfasst eine Kreditkarten-Transaktion [KEF+98]:

- Der Halter einer Kreditkarte nutzt diese, um eine bestimmte Ware eines Verkäufers zu erwerben.
- Der Verkäufer übermittelt am Ende des Tages elektronisch die Zahlungsanweisung des Kunden zusammen mit allen übrigen zu Kreditkarten korrespondierenden Zahlungsanweisung zu seiner Bank.
- Die Bank des Verkäufers schreibt diesem den ihm zustehenden Betrag gut. Dann versendet die Bank des Verkäufers die Zahlungsanweisung des Kartenhalters, sowie die übrigen vom Verkäufer erhaltenen Zahlungsanweisungen, an die die Kreditkarte ausstellende Gesellschaft.
- Diese Gesellschaft leitet die Zahlungsaufträge zu den Banken der Karteninhaber weiter.
- Dann errechnet diese Gesellschaft den Betrag, den die Banken der Karteninhaber ihr schuldig sind, auch die Bank des Karteninhabers.
- Die Banken der Kreditkarteninhaber überweisen diese Beträge via Fedwire.
- Die Kreditkarten ausstellende Gesellschaft bezahlt die Bank des Verkäufers
- Dann stellt diese Gesellschaft den entsprechenden Betrag dem Karteninhaber in Rechnung.
- Die Bank des Karteninhabers überweist den genannten Betrag an die Gesellschaft, die ihm die Karte ausgegeben hat.

Fazit. Diese und andere auf computergestützten Informationssystemen fußenden Bezahlsysteme sind essenziell für den reibungslosen Ablauf von bargeldlosen Transaktionen und damit für das Funktionieren der Volkswirtschaften hoch entwickelter Länder. Sie müssen für einen verlässlichen, fehlerfreien und pünktlichen Geldtransfer garantieren, die getätigten Transaktionen müssen hinreichend sicher sein und deren Endgültigkeit sowie Unumkehrbarkeit muss gewährleistet werden können. Die beschriebenen Systeme beugen dabei betrieblichen Risiken, wie Störungen und Ausfällen einzelner Komponenten, mit Backup-Einrichtungen und automatisierten Recovery-Maßnahmen vor. Den Risiken, die von Personen mit betrügerischer Absicht ausgehen, wird mit Zugriffskontrollen, Authentisierungsmaßnahmen und Verschlüsselungstechniken vorgebeugt.

2.2 Das Schienenverkehrssystem der USA

Das Schienenverkehrswesen ist einer der Eckpfeiler der US-Volkswirtschaft. Ein Großteil der Rohstoffe für das produzierende Gewerbe, wie Kohle, Holz und Chemikalien wird von Eisenbahn-Transportunternehmen zu Elektrizitätswerken, Raffinerien und Automobilwerken transportiert, deren fertige Produkte dann auf der Schiene weiter in die USA und zum Rest der Welt verfrachtet werden.

Die Infrastruktur des Schienenverkehrssystems ist weiträumig verteilt. Die Eisenbahnunternehmen kontrollieren ihre Operationen mithilfe mehrerer autonomer Informationssysteme, die in einem zentralen Kontrollzentrum föderiert werden. Ihre teils proprietären, teils mitbenutzten öffentlichen Informationsnetzwerke liefern in

Echtzeit die Informationen, welche benötigt werden, um ein holistisches Bild des Zugverkehrssystems zu erlangen. Die eingehenden Informationen ermöglichen es den Weichenstellern und Zugüberwachern im Kontrollzentrum, tausende Züge am Tag zu überwachen und diese richtig zu delegieren. Ad hoc bieten die Systeme unter anderem folgendes an:

- Ein Ortsbestimmungssystem, das die Züge präzise orten kann.
- Ein System, mit welchem man zentral die Eisenbahnweichen verstellen kann.
- Ein Kommunikationssystem zur Verständigung mit den Lokführern, unter anderem um Anweisungen bezüglich der Geschwindigkeit der Züge erteilen zu können.
- Ein System zum automatischen Überwachen der Lokomotiven, um Wartungsprobleme zu antizipieren und Ausfälle vermeiden zu können.

Systeme und Betreiber. Es gibt derzeit rund 200 Eisenbahngesellschaften, die alle ihre eigenen Informationssysteme entwickelt haben und nutzen. Zur Administration der systemweiten Informations- und Kommunikationsnetzwerke, zum Einführen und Setzen von Standards für das Schienenverkehrswesen und zum Betreiben weiterer Forschung auf diesen Gebieten haben die größeren der Frachtbeförderungsunternehmen die Association of *American Railroads (AAR)* ins Leben gerufen. Jede Eisenbahngesellschaft nutzt ihr eigenes System und nutzt dabei einen Teil oder das gesamte AAR-Informationsnetzwerk. Obwohl es viele Gemeinsamkeiten zwischen den Systemen gibt, entwickelten sich alle unterschiedlich. Um diese unterschiedlichen Systeme zu verknüpfen und Informationen reibungslos austauschen zu können, wurde das Kommunikationsnetzwerk namens *RAILINC* entwickelt.
Eines der wichtigsten Systeme, das von der AAR administriert wird, ist das *Telerail Automated Information Network (TRAIN II)*, das Daten von Frachtwaggons, Containern etc., sammelt. Mehr als 200 Gesellschaften speisen mehr als 3 Millionen Eingaben pro Tag in das System ein, beispielsweise Abfahrts- und Ankunftszeiten und Frachtbriefe. Train II verarbeitet diese Daten und stellt diese den zentralen Stellen zum Überwachen der Lieferungen, zur Rechnungskontrolle für an andere Gesellschaften verliehene Schienenfahrzeuge und Ähnlichem bereit. Train II besteht aus diversen automatisierten Subsystemen, wobei die AAR als zentrale Informationssammelstelle fungiert. Die wichtigsten Teilsysteme sind:

- *Interline Trace*: Dieses System bietet den Gesellschaften die Möglichkeit zum automatisierten Überwachen verschiedener Frachtstücke an, beispielsweise mithilfe der in den Frachtbriefen vermerkten Ortsangaben.
- *Car Location Messages (CLMs)*: Mithilfe von Scannern, die Barkodes lesen, welche jedem Schienenfahrzeug zugewiesen wurden, kann deren Position bestimmt werden. Die entsprechenden Meldungen werden in den Host-Rechner eingespeist und vom System gesammelt.
- *Electronic Data Exchange (EDE)*: EDE, welches von der AAR geleitet wird, ist eines der größten Netzwerke der Welt, welches auf der Electronic Data Interchange Technologie basiert; Pro Tag werden mehr als eine Mil-

lion Nachrichten zwischen Handelspartnern versandt. Typische Meldungen sind dabei Frachtbriefe, Rechnungen und Kaufaufträge.

- *Interline Settlement System (ISS)*: Dieses System bietet den Eisenbahngesellschaften die Möglichkeit, untereinander ihre Rechnungen zu begleichen, beispielsweise für das Verleihen von Schienenfahrzeugen, für Frachtverlust oder -beschädigung und das Einziehen oder Bezahlen eines entsprechendes für Betrags das Befördern von Frachtgütern.

Fazit. Das System wird sich künftig dahingehend entwickeln, dass es mehr und mehr automatisiert werden wird und sich menschliche Interaktion auf die bloße Überwachung des Systems beschränken wird; Diese Entwicklung wird beispielsweise mit der geplanten Erweiterung des Systems um das so genannte *Positive Train Separation (PTS)* Kontrollsystem vorangetrieben, ein satellitengestütztes System, das ohne menschliche Beteiligung die Bewegungen sämtlicher Schienenfahrzeuge kontrollieren und Kollisionen vermeiden soll. Des Weiteren sollen ausfallsichere Systeme in Zukunft die vollautomatisierte Kontrolle des Schienenverkehrssystems Realität werden lassen, welche permanent ihren Dienst verrichten, ungeachtet dessen, wie viele Schienenfahrzeuge gerade in Aktion sind.

Weitere Beispiele für auf komplexen computergestützten Informationssystemen beruhenden kritischen Infrastruktur-Einrichtungen, wie Luftraumüberwachung, Stromversorgung oder Telekommunikationseinrichtungen lassen sich leicht finden. Die Bedeutung solcher kritischen Infrastruktur-Einrichtungen zeigt die Überlegung, dass das normale Leben in Wirtschaft und Gesellschaft schwer beeinträchtigt wäre, würden die zugrunde liegenden Informationssysteme ihre Funktion auch nur kurzzeitig verweigern oder fehlerhaft arbeiten, was durch die Interdependenzen zwischen diesen System noch verschlimmert wird; So würde beispielsweise der Ausfall der Stromversorgung eine Beeinträchtigung oder gar den Ausfall der übrigen genannten Systeme bedingen. Auch andere Informationssysteme, deren Ausfall das öffentliche Leben nicht dermaßen beeinträchtigen und solch große volkswirtschaftliche Schäden verursachen können wie die oben genannten, sind kritische Komponenten für die Unternehmen, die sie betreiben, beispielsweise im Internethandelhandel, Onlinebanking etc. Ein auch nur kurzeitiges Nichtverfügbarsein kann den Unternehmen immensen finanziellen Schaden zufügen, fehlerhaftes Arbeiten oder Sicherheitslücken das Vertrauen der Kundschaft untergraben etc.

3 Anforderungen an Informationssysteme

Die Ansprüche an solche Systeme, vor allem deren Verlässlichkeit und Fähigkeit der Anpassung an sich häufig ändernde Bedingungen betreffend, sind aufgrund dessen außergewöhnlich hoch. Denn viele Systeme arbeiten auf landesweit implementierten oder gar weltumspannenden Netzwerken und müssen 24 Stunden am Tag und 365 Tage im Jahr funktionieren, sie müssen in hohem Maße verfügbar sein, netzwerkweite Sicherheit bieten und ohne große Schwierigkeiten erweiterbar sein. Turing-Award-Gewinner Jim Gray hat wünschenswerte „Fernziele“ in drei zentralen Punk-

ten zusammengefasst [Gray99]:

Das System muss *störungsfrei* arbeiten
Es sollte von einer Vielzahl von Personen benutzt werden können, der Aufwand für dessen Administration sollte aber minimal sein. Ziel dabei ist, dass der System-Manager lediglich die Zielvorgaben setzt, das System sollte dann den Rest tun. Die Arbeit sollte auf viele Komponenten verteilt werden. Das System sollte ohne Schwierigkeiten erweitert werden können: Wenn neue Module dazukommen sollen, sollte man diese einfach „einstöpseln“ können. Wenn eine Komponente fehlerhaft arbeitet oder den Dienst versagt, soll dessen Funktion von anderen Teilen des Systems übernommen werden. Das System sollte selbst imstande sein, einen Hardwarefehler zu erkennen und automatisch die entsprechenden Ersatzteile per Eilpost schicken lassen. Hard- und Softwareupgrades sollten automatisch vonstatten gehen.

Das System muss *sicher* sein
Dabei muss garantiert werden können, dass nur autorisierte Nutzer den Dienst des Systems in Anspruch nehmen können und dass nicht autorisierte Personen weder autorisierten Nutzern der Zugang zum System verwehren noch an Informationen aus dem System gelangen und diese löschen oder missbrauchen können. Eine hinreichend sichere korrekte Authentifizierung scheint dabei aufgrund der Möglichkeit des Ausspähens von Passworten und PINs nur über die Identifikation biometrisch eindeutiger und fälschungssicherer Merkmale möglich. Mindestanforderung für die Implementierung eines sicheren Systems ist die, dass kein „Tiger-Team“ imstande ist, in das System einzudringen; Da dies aber de facto keine hundertprozentige Sicherheit für das System garantieren kann, da niemals alle denk-baren bzw. möglichen Angriffe auf das System auf deren Erfolgsmöglichkeiten geprüft werden können, bedarf es eines Beweises für die Unverwundbarkeit des Systems gegenüber böswilligen Attacken.

Das System soll permanent *verfügbar* sein
Dabei soll garantiert werden können, dass das System in hundert Jahren nicht mehr als eine Sekunde nicht verfügbar ist. Um dieses Ziel zu erreichen, bedarf es der Bereitstellung einer großen Bandbreite für Übertragungen und eines großen Hardwareaufwandes, um bestimmte kritische Komponenten an verschiedenen entfernten Orten zu duplizieren. Des Weiteren müssen Transaktionen verwendet werden, um die Konsistenz gespeicherter Daten nicht zu gefährden. Da Fehler auch beim Zusammenspiel verschiedener Komponenten auftreten können, sollten zudem bei der Schaffung von Redundanz, für Notfalldienst und Recovery-Maßnahmen, von verschiedenen Entwurfsprinzipien bei deren Realisierung Gebrauch gemacht werden. Beschädigte Komponenten sollten möglichst schnell repariert werden. Das Erfülltsein dieser Bedingung kann nicht durch bloßes Testen gezeigt werden; Auch hier bedarf es eines Beweises.

4 Adaptierung, Zuverlässigkeit und Sicherheit

Informationssysteme, welche die drei oben genannten Eigenschaften erfüllen, sind so genannte *zuverlässige adaptive Informationssysteme* (engl.: dependable adaptive information systems, *DAIS*). Zur detaillierteren Diskussion der Anforderungen bei Entwurf und Implementierung von DAIS werden an dieser Stelle zunächst die zentralen Termini in diesem Zusammenhang, nämlich Adaptivität, Zuverlässigkeit und Sicherheit voneinander abgegrenzt und erläutert werden. Die Begriffe werden dabei allgemein, also nicht auf Informationssysteme zugeschnitten, definiert und erklärt werden.

4.1 Grundlegende Begriffe

Als Dienst (engl.: service) eines Systems wird dessen vom Benutzer wahrgenommenes Verhalten bezeichnet; Benutzer kann dabei sowohl eine Person als auch ein anderes System sein. Als externer Status wird der Teil des gegenwärtigen Gesamtzustandes des Systems bezeichnet, der vom Benutzer an dessen Dienst-Schnittstelle wahrgenommen werden kann; Der verbleibende Teil des Gesamtzustandes ist der interne Status. Damit ist dann der vom System betriebene Dienst eine Folge externer Zustände.

4.2 Adaptierung

Der Begriff *Adaptierung* ist ein Oberbegriff für alle Konzepte zur automatischen Anpassung eines Systems an sich ändernde Anforderungen und Einwirkungen auf das System. Dabei sollen etwaige Anpassungen bestimmter Systemparameter vom System selbst vollzogen werden, die Systeme sollten also *selbst-verwaltend* (engl.: self-managing) sein. Die Tätigkeit solcher Systeme wird in der Literatur als *autonomes Rechnen* (engl.: autonomic computing) bezeichnet.
Die Intention autonomen Rechnens macht eine Analogie mit dem autonomen Nervensytem des Menschen deutlich: Das autonome Nervensytem entlastet unser Bewusstsein von der Aufgabe, sich mit Funktionen niederer Ebenen, aber lebenswichtigen Funktionen, des Körpers befassen zu müssen. Gleichermaßen entlastet autonomes Rechnen die Administratoren von Systemen von einigen ihrer – wichtigen – operativen Aufgaben [GaCo03].

Notwendigkeit von autonomem Rechnen. Nachdem die Computerindustrie Jahrzehnte damit zugebracht hat, große und immerwachsende Systeme zu schaffen, ist heute deren Umfang und Komplexität ein Problem geworden. Denn das Testen und die Leistungsverbesserung (engl.: tuning) von komplexen, oft heterogenen Systemen, bestehend aus Dutzenden von Anwendungen, hunderten von Systemkomponenten und tausenden von Leistungsverbesserung-Parametern, wird zusehends schwieriger und das Administrieren einer wachsenden Zahl von Details im System-Management ist sehr arbeitsintensiv.

Dies spiegelt sich auch in den folgenden Schätzungen und Studien wider [GaCo03]:

- Es wird geschätzt, dass zwischen einem Drittel und der Hälfte des IT-Budgets eines Unternehmens ausgegeben werden, um Systemausfällen vorzubeugen, bzw. diese nach einem Ausfall wieder herzustellen.
- Eine weitere Schätzung besagt, dass 40 Prozent der Software-Arbeit einer Gruppe zum Testen benötigt wird; Wächst der Umfang des Sytemverhaltens, so wächst der Umfang der Test-Software im Vergleich dazu exponentiell.
- Eine allgemeine Faustregel ist: Für jeden Dollar, mit dem Speicherplatz gekauft wird, müssen neun weitere Dollars investiert werden, um jemanden zu haben, der diesen verwaltet.
- Eine Studie der Aberdeen-Gruppe besagt, dass die administrativen Kosten 60 bis 75 Prozent der Gesamtkosten betragen, die für den Betrieb einer Datenbank anfallen; Diese Kosten umfassen unter anderem administrative Tools, Installation, Upgrades, Training, Gehälter der Administratoren sowie Service und Betreuung vom Hersteller der Datenbank.

All dies führt zu dem Schluss, dass das System-Management komplexer Systeme vereinfacht und automatisiert werden muss. Irving Wladawsky-Berger formuliert diese Erkenntnis so [GaCo03]: „Es gibt nur eine Antwort: Das System muss sich selbst verwalten. Damit meine ich kein weit hergeholtes AI-Projekt – wovon ich spreche, ist die Notwendigkeit, die richtige Software zu entwickeln, die richtige Architektur, die richtigen Mechanismen, [...] Anstatt also sich wie gewöhnlich zu verhalten und eine Person zu benötigen für alles, was es tut, sollte das System beginnen, sich zu verhalten, wie wir es von einem 'intelligenten' Computer erwarten würden und sich um seine eigenen Bedürfnisse kümmern. Wenn es sich nicht wohl fühlt, tut es etwas dagegen. Wenn es jemand angreift, so bemerkt es das System und befasst sich mit dem Angriff. Wenn es mehr Rechenleistung benötigt, so holt es sich diese und wartet nicht länger erst auf einen Menschen, der dies tut“.

Eigenschaften von autonomem Rechnen. Autonom rechnende Systeme haben folgende Charakteristika [GaCo03]:

- Um autonom zu sein, muss das System „sich selbst kennen“ – und aus Komponenten bestehen, die auch eine System-Identität haben.
- Ein autonomes System muss sich selbst unter verschiedenen und nicht vorhersagbaren Bedingungen konfigurieren und rekonfigurieren.
- Ein autonomes System verbleibt nicht im Status quo, es sucht stets nach Möglichkeiten, seinen Betrieb zu optimieren.
- Ein autonomes System muss etwas Ähnliches wie Heilung vollziehen – es muss imstande sein, sich von Routine- und außergewöhnlichen Ereignissen, die in einigen Komponenten eine Fehlfunktion auslösen, zu erholen.
- Eine virtuelle Welt ist nicht weniger gefährlich als die reale, deswegen muss ein autonomes System ein Experte im Selbst-Schutz sein.

- Ein autonomes System kennt sein Umfeld und den Kontext, der seine Aktivität umfasst, und handelt dementsprechend.
- Ein autonomes System kann nicht in einem abgeschlossenen Umfeld existieren, es muss offenen Standards genügen.
- Die vielleicht wichtigste Eigenschaft für den Benutzer: Ein autonomes System antizipiert optimal die Ressourcen, die benötigt werden, um dem Benutzer die geforderte Information zu liefern, wobei es seine Komplexität vor ihm verbirgt.

Lässt man diese Charakteristika den Begriff „selbstverwaltende Systeme“ definieren, so werden künftige autonome Systeme über die folgenden vier fundamentalen Fähigkeiten verfügen [GaCo03]:

- Das System muss *selbstkonfigurierend* (engl.: self-configuring) sein: Das System passt sich automatisch an das sich dynamisch verändernde Umfeld an. Dieser Aspekt der Selbstverwaltung bedeutet, dass neue Features, Software und Server dynamisch zu Systemressourcen hinzugefügt werden, ohne Unterbrechung des Dienstes. Das System muss so entworfen werden, um diesen Aspekt auf einer abstrakten Ebene zu bieten, durch Möglichkeiten wie Plug-and-Play-Bausteine, Konfigurations- und Setup-Wizards und kabelloses Server-Management. Diese Features erlauben es, dass Funktionen dynamisch zu Systemressourcen mit einem Minimum an menschlicher Intervention hinzugefügt werden. Selbst-Konfiguration beinhaltet nicht nur die Fähigkeit für ein einzelnes System, sich selbst dynamisch zu konfigurieren, sondern auch für mehrere Systeme innerhalb eines Unternehmens, sich beispielsweise zu einer E-Business-Infrastruktur des Unternehmens zu gruppieren. Das Ziel autonomen Rechnens ist es, die Fähigkeit zur Selbst-Konfiguration für die ganze IT-Infrastruktur zu bieten, nicht nur für einzelne Server, Software und Speicherbausteine.
- Das System muss *selbstheilend* (engl.: self-healing) sein: Das System entdeckt, diagnostiziert Störungen und reagiert auf diese. Damit ein System selbst-heilend ist, muss es imstande sein, sich von einer beschädigten Komponente zu „erholen“, indem es deren Beschädigung erkennt und die Komponente isoliert, diese offline setzt, deren Reparatur oder Auswechslung veranlasst und anschließend diese Komponente wieder in Betrieb nimmt – und das alles ohne merkliche Unterbrechung des eigenen Dienstes. Das System muss Probleme vorhersehen und Vorsorge treffen können, dass ein Fehler keine Auswirkung auf die Anwendung hat. Die Zielvorstellung selbst-heilender Systeme ist es, Systemausfälle zu minimieren um die Unternehmensanwendung stets verfügbar zu halten.
- Das System muss *selbstoptimierend* (engl.: self-optimizing) sein: Das System überwacht seine Ressourcen und stimmt diese automatisch richtig ab. Selbst-Optimierung betrifft Hard- und Software, um effizient den Nutzen der Ressourcen zu maximieren und die Anforderungen der Endnutzer ohne menschliche Intervention optimal erfüllen zu können. Diese Fähigkeit sollte

erweitert werden auf mehrere heterogene Systeme eines Unternehmens, um eine Sammlung von Ressourcen zu bieten, die unternehmensweit von einem „logischen“ Workload-Manager verwaltet werden könnten. Ressourcen-Allokation und Workload-Management müssen dynamisches Umverteilen von Workloads zu solchen Systemen erlauben, die über notwendigen Ressourcen verfügen, um die Anforderungen des Workloads zu erfüllen. Auf ähnliche Weise müssen Speicherbausteine, Datenbanken, Netzwerke und andere Ressourcen kontinuierlich hinsichtlich ihrer Leistung verbessert (engl.: tuned) werden, um effiziente Operationen selbst unter unvorhersagbaren Bedingungen zu ermöglichen. Diese Features erlauben es einem Unternehmen, das Nutzen seiner Ressourcen zu optimieren.

- Das System muss *selbstschützend* (engl.: self-protecting) sein: Das System antizipiert, erkennt, identifiziert Angriffe und schützt sich vor diesen. Selbst-schützende Systeme müssen die Fähigkeit haben, Benutzerzugriffe auf alle Ressourcen des Systems durchzuführen, sich aber vor unautorisiertem Systemzugriff zu schützen. Sie müssen Versuche des Eindringens in das System bemerken, melden, diese Aktivitäten unterbinden und Back-up- und Recovery-Einrichtungen bieten, die ebenso sicher sind wie das Originalsystem.

Künftige Entwicklung. Heutige Hard- und Softwaresysteme zeigen schon Tendenzen, sich Richtung autonomes Verhalten zu entwickeln:

- Neue Software-Tools von Datenbanken können deren Statistiken nutzen, diese analysieren und von der Information über zurückliegende Leistung lernen.
- Webserver können in Echtzeit Informationen einer Selbstdiagnose liefern. Dies ermöglicht es den Kunden, Probleme mit Ressourcen schneller zu erkennen, anstatt auf eine Meldung nach dem Vorfall zu warten, die das Problem meldet.
- Server können Software-Algorithmen nutzen, welche die Muster der Internetzugriffe (beispielsweise Montag-Morgen-Peak) analysieren und ihre Ressourcen dementsprechend bereitstellen.

Es gibt schon Fortschritte auf diesem Feld, aber es gibt auch noch viel zu tun; So müssen die Bemühungen, ein zusammenwirkendes Systemverhalten zu erreichen, weit über die Verbesserungen einzelner Komponenten hinausgehen. Diese Komponenten müssen föderiert werden, eine integrierte Architektur bilden, sodass Gruppen von Komponenten zusammenarbeiten können. System-Management-Werkzeuge spielen dabei für die Systemadministration durch das Koordinieren der Aktionen der einzelnen Komponenten und durch das Bereitstellen eines vereinfachten Mechanismus eine zentrale Rolle. Da die meisten IT-Infrastrukturen aus Komponenten verschiedener Anbieter bestehen, sind offene Industriestandards der Schlüssel zu Konstruktion solcher Systeme mit autonomen Subsystemen. Diese Standards

ermöglichen das Kommunizieren autonomer Systemelemente, die sich ihrerseits wieder selbst verwalten, sich also intern selbst konfigurieren, nach Möglichkeit selbst heilen, sich selbst optimieren und selbst schützen vor externen Angriffen. Autonome Elemente bilden auf diese Weise Blöcke, die ein autonomes System schaffen.
Autonome Elemente überwachen permanent das System- oder Komponentenverhalten durch „Sensoren“ und vollziehen Adaptierungen durch „Effektoren“. Durch das Überwachen des Verhaltens mithilfe der Sensoren, das Analysieren deren Daten, und das Vorausplanen, welche Aktion als nächstes auszuführen ist, wenn denn eine Aktion vonnöten ist, und das Ausführen durch Effektoren, wird eine Art *Onlinefeedback-Kontrollschleife* geschaffen.

Untereinander verbundene autonome Elemente benötigen Mechanismen verteilter Rechensysteme, um auf Ressourcen in dem Netzwerk zuzugreifen. Der Begriff *Grid Computing* bezeichnet die Idee einer Infrastruktur, die durch das Vernetzen heterogener verteilter Computersysteme gebildet wird. Die Entwicklung zum Grid Computing wird vielfach als der nächste evolutorische Schritt des Internet gesehen. Der Begriff „Grid“ – Netz – wurde gewählt in Analogie zum Stromnetz, das ubiquitären Zugriff auf die Ressource Strom gewährleistet.

Autonomes Rechnen stellt eine relativ neue Richtung in der Forschung Informatik dar. Die große Herausforderung auf diesem Feld wird Forscher diverser Bereiche der Informatik beschäftigen, Bereiche wie System-Management, verteiltes Rechnen, Netzwerke, Softwareentwicklung, künstliche Intelligenz und andere mehr [GaCo03].

4.3 Zuverlässigkeit und Sicherheit

In der Literatur sind die Begriffe *Zuverlässigkeit* (engl.: dependability) und *Sicherheit* (engl.: security) häufig so definiert, dass Sicherheit nur als ein Aspekt von verlässlichen Systemen charakterisiert ist. Der besseren Übersicht wegen wird Sicherheit hier aber als eigenständiger und gleichrangiger Begriff vorgestellt werden.

Definition von Zuverlässigkeit. Die ursprüngliche Definition für die Zuverlässigkeit eines Systems ist seine Fähigkeit, seinen Dienst dergestalt zu verrichten, dass man dieser Tatsache gerechtfertigter Weise trauen kann. Diese Definition betont die Rechtfertigung für das in das System gesetzte *Vertrauen* (engl.: trust). Man sagt gewöhnlich, die Zuverlässigkeit eines Systems muss die Zuverlässigkeit eines solchen Systems nicht übersteigen, von dessen Funktionieren es abhängt. Die Abhängigkeit eines Systems A von einem System B repräsentiert also das Ausmaß, mit dem die Zuverlässigkeit von B diejenige von A beeinflusst; Sie kann von völliger Abhängigkeit (der Ausfall von System B würde den Ausfall von System A nach sich ziehen) bis zur vollständigen Unabhängigkeit reichen (System B kann nicht Ursache eines Ausfalls von System A sein). Mithilfe dieses Konzepts der Abhängigkeit lässt sich nun Vertrauen definieren als Grad der akzeptierten Abhängigkeit [ALR+04].

Eine alternative Definition für Zuverlässigkeit eines Systems, mit Fokus auf der Tatsache, dass es seinen Dienst zuverlässig ausführen kann, ist dessen Fähigkeit, Dienstfehlfunktionen in dem Maße zu vermeiden, dass diese nicht schwerer und häufiger sind als akzeptabel [ALR+04].

Zuverlässigkeit und seine Attribute. Der Begriff Zuverlässigkeit ist definiert als Kombination folgender Attribute [ALR+04]:

- *Verfügbarkeit* (engl.: availability): Das Bereitsein des Systems, seinen Dienst richtig zu verrichten.
- *Ausfallsicherheit* (engl.: reliability): Die Stetigkeit des Systems beim korrekten Ausführen seines Dienstes.
- *Betriebssicherheit* (engl.: safety): Weder Nutzer und noch Umfeld erleiden Schaden durch das System.
- *Integrität* (engl.: integrity): Das Ausbleiben unsachgemäßer Änderungen am System.
- *Wartbarkeit* (engl.: maintainability): Die Fähigkeit, Reparaturen und Modifikationen zu überstehen.

Sicherheit und seine Attribute. Der Begriff Sicherheit ist definiert als Kombination folgender Attribute [ALR+04]:

- Verfügbarkeit: Das Bereitsein, korrekten Dienst zu verrichten, mit der Einschränkung, dass nur autorisierte Operationen vollzogen werden.
- Integrität: Das Ausbleiben unautorisierter Änderungen am System.
- *Vertraulichkeit* (engl.: confidentiality): Das Unterbleiben der Preisgabe von Informationen an nicht autorisierte Nutzer.

Abbildung 1 illustriert nochmals die beiden Begriffe als Kombination ihrer korrespondierenden Attribute.

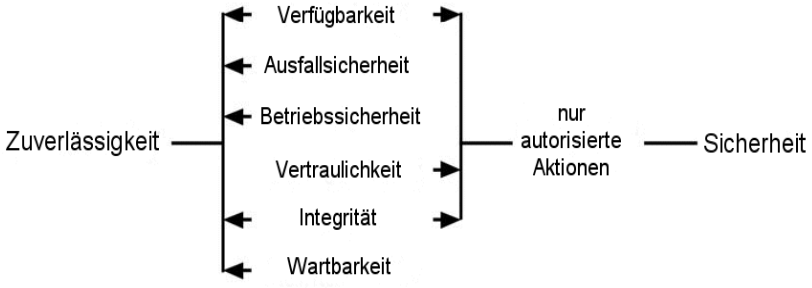

Abbildung 1. Zuverlässigkeit und Sicherheit mit ihren Attributen

Der Grad, zu dem ein System die Attribute von Zuverlässigkeit und Sicherheit erfüllt, sollte nicht absolut determiniert gesehen werden: Da Fehler generell nicht zu vermeiden sind, können Systeme nach menschlichem Ermessen niemals völlig verfügbar, ausfallsicher oder betriebssicher sein – ein Fakt, der Jim Gray's Forderung nach Beweisen die Zuverlässigkeit eines Systems betreffend in weite Ferne rücken lässt, wenn nicht gar ad absurdum führt.

Sekundärattribute. Neben den diskutierten Primärattributen von Zuverlässigkeit und Sicherheit können weitere sekundäre Attribute definiert werden, welche die primären Attribute verfeinern oder spezialisieren. Beispielsweise charakterisiert *Robustheit* (engl.: robustness) die Zuverlässigkeit eines Systems hinsichtlich dessen Umgang mit externen Systemfehlern.
Sekundäre Attribute sind im Besonderen wichtig, um verschiedene Arten von Informationen die Sicherheit eines Systems betreffend zu unterscheiden. Exemplarisch dafür seien die folgenden Sekundärattribute genannt [ALR+04]:

- *Verantwortlichkeit* (engl.: accountability): Vorhandensein und Vollständigkeit der Identität derjenigen Person, welche eine Operation am System vollzieht.
- *Authentizität* (engl.: authenticity): Die Vollständigkeit einer Nachricht bezüglich deren Inhalt und Herkunft und etwaiger weiterer Informationen, beispielsweise der Zeitpunkt ihres Verschickens.
- *Verbindlichkeit* (engl.: nonrepudiability): Vorhandensein und Vollständigkeit der Identität des Absenders einer Nachricht (Verbindlichkeit der Herkunft) oder des Empfängers (Verbindlichkeit des Erhalts).

Ähnliche Konzepte. Es existieren weitere Konzepte ähnlich dem der Zuverlässigkeit. Sie sind, was die Ziele betrifft äquivalent, beispielsweise [ALR+04]:

- Überlebensfähigkeit (engl.: survivability): Das System ist imstande, seinen Auftrag innerhalb akzeptabler zeitlicher Parameter zu erfüllen, selbst wenn es angegriffen wird oder Fehler auftreten. Überlebensfähigkeit geht über Sicherheit und Fehlertoleranz hinaus und zielt dabei stark auf das Ausführen grundlegender Dienste ab.
- Vertrauenswürdigkeit (engl.: trustworthiness): Der Grad an Vertrauen, das in das System und seine Fähigkeit, wie erwartet zu arbeiten, gesetzt wird.

4.4 Gefährdung von Zuverlässigkeit und Sicherheit

Das System verrichtet seinen Dienst korrekt, wenn es den Anforderungen seiner funktionalen Spezifikation entspricht; Anderenfalls tritt eine *Dienstfehlfunktion*, kurz *Fehlfunktion* (engl.: failure) auf, das heißt entweder, dass das System nicht seinen funktionalen Anforderungen entspricht oder diese das System unvollständig beschreiben. Der Zeitraum, während dem das System seinen Dienst nicht korrekt verrichtet, heißt *Dienst-Ausfall* (engl.: service outage), der Übergang vom inkorrekten zum korrekten Dienst heißt *Dienst-Wiederherstellung* (engl.: service restoration).

Da der Dienst eines Systems eine Folge externer Zustände ist, impliziert eine Dienstfehlfunktion, dass zumindest ein externer Status des Systems von der Spezifikation des korrekten Dienst abweicht; Ein solches Abweichen wird als *Fehlverhalten* (engl.: error) bezeichnet. Die einem Fehlverhalten zugrunde liegende Ursache ist ein *Fehler* (engl.: fault). Ein Fehler heißt aktiv, wenn er ein Fehlverhalten verursacht, andernfalls latent; er kann systemintern oder -extern sein. Das Vorliegen von Schadensanfälligkeit (engl.: vulnerability), einem internen Fehler, ist notwendig, damit ein externer Fehler, durch Auslösen eines Fehlverhaltens und möglicherweise nachfolgenden Fehlfunktionen, das System schädigen kann. Umfasst die Spezifikation eines Systems mehrere Funktionen, so kann die Fehlfunktion eines oder mehrerer Dienste das System in einen Zustand verminderter Leistungsfähigkeit versetzen, wobei nur noch ein Teil der Dienstes dem Benutzer zur Verfügung steht. Die System-Spezifikation kann mehrere derartiger Modi beschreiben, beispielsweise langsamer Dienst, einge-schränkter Dienst, Notfalldienst, etc. Hier spricht man von einer teilweisen Fehlfunktion des Systems, dessen Funktionalität oder Leistungsfähigkeit betreffend.

Fehler. Während seiner Entwicklungsphase interagiert das System mit seiner Entwicklungsumgebung. Von dieser Entwicklungsumgebung können Entwicklungsfehler in das System eingeschleust werden. Die Entwicklungsumgebung besteht aus folgenden Elementen und Agenten [ALR+04]:

- Die physische Welt mit ihren natürlichen Phänomenen.
- Das System entwickelnde Personen, möglicherweise mit mangelnder Kompetenz oder böswilligen Absichten.
- Entwicklungswerkzeuge: Soft- und Hardware, die von den Entwicklern während des Entwicklungsprozesses zu Hilfe genommen werden.
- Produktions- und Testanlagen.

Während seiner Nutzungsphase interagiert das System mit seinem Benutzerumfeld und könnte von inhärenten Fehlern betroffen sein. Das Benutzerumfeld besteht aus den folgenden Elementen und Agenten [ALR+04]:

- Die physische Welt mit ihren natürlichen Phänomenen.
- Administratoren und Wartungspersonal: Personen oder andere Systeme, die autorisiert sind, das System zu verwalten, modifizieren, reparieren und zu benutzen. Ihnen könnte es an Kompetenz mangeln und sie könnten böswillige Absichten hegen.
- Benutzer: Personen oder andere Systeme, welchen den Dienst des Systems an dessen Dienstschnittstelle nutzen.
- Provider: Personen oder andere Systeme, deren Dienst das betrachtete System nutzt.
- Infrastruktureinrichtungen: Funktionseinheiten, die dem System spezielle Dienste anbieten, wie Informationsquellen (beispielsweise Uhren, GPS), Kommunikationseinrichtungen, Energiequellen, Kühlsysteme, etc.
- Eindringlinge (engl.: intruders): Bösartige Personen oder Systeme, die

versuchen, den Dienst des Systems zu ändern oder dessen Verrichtung durch das System vollständig zu unterbinden, die Funktionalität oder Leistungsfähigkeit des Systems zu verändern oder Zugriff auf vertraulichen Informationen zu erlangen, wozu sie nicht autorisiert sind. Dies sind beispielsweise Hacker, korrumpierte Insider, feindliche Regierungen oder Organisationen und bösartige Software.

Alle Fehler, welche ein System betreffen könnten, können in drei große Gruppen eingeteilt werden, die sich teilweise überschneiden, nämlich [ALR+04]:

- *Entwicklungsfehler:* Betrifft alle Fehler, die während seiner Entwicklung in das System eingeschleust werden.
- *Physische Fehler:* Alle Fehler, welche die Hardware des Systems betreffen.
- *Interaktionsfehler:* Betrifft alle systemexternen Fehler des Systems, die während dessen Nutzungsphase auftreten.

Fehlfunktionen. Fehlfunktionen lassen sich auf dreierlei Weise charakterisieren; Auch hier überschneiden sich die Begriffe etwas [ALR+04]:

- Oben genannte Entwicklungsfehler können eine *Entwicklungsfehlfunktion* verursachen, beispielsweise die Verminderung der Leistungsfähigkeit des Systems infolge von Einschränkungen am System wegen eines zu knapp veranschlagten Budgets, wegen unvollständiger oder fehlerhafter Spezifikation etc.
- Eine oben schon genannte *Dienstfehlfunktion* ist definiert als das Sze-nario, wenn der vom System verrichtete Dienst vom korrekten, in der Spezifikation des Systems festgehaltenen Dienst abweicht.
- Eine *Zuverlässigkeits-* bzw. *Sicherheitsfehlfunktion* tritt dann auf, wenn das System öfter oder schwerere Dienstfehlfunktionen aufweist, als in der Spezifikation des Systems als akzeptabel beschrieben.

4.5 Konzepte zum Erreichen von Zuverlässigkeit und Sicherheit

Bei der Implementierung zuverlässiger und sicherer Systeme beeinflusst der Grad der Betonung der einzelnen Attribute direkt die Balance der Techniken, die zum Erreichen der Vorgaben eingesetzt werden. Die Konzepte solcher Techniken zur Umsetzung der verschiedenen Attribute von Zuverlässigkeit und Sicherheit können in vier Kategorien unterteilt werden [ALR+04]:

- *Fehlerprävention*: Vorbeugen vor dem Auftreten oder Einschleusen von Fehlern.
- *Fehlertoleranz*: Vermeiden von Dienstfehlfunktionen, wenn Fehler auftreten.
- *Fehlerbeseitigung*: Reduzieren der Anzahl und Verminderung ernsthafter Auswirkungen von Fehlern.
- *Fehlervorhersage*: Abschätzen der gegenwärtige Anzahl, des künftigen Auftretens und der wahrscheinlichen Konsequenzen von Fehlern.

Dabei ist es Ziel von Fehlertoleranz und Fehlerprävention, das Vertrauen der Benutzer in die Fähigkeit eines Systems steigern, seinen Dienst korrekt zu verrichten, wogegen Fehlerbeseitigung und Fehlervorhersage darauf abzielen, das Vertrauen in die Fähigkeit des Systems zu steigern, dass seine funktionalen Spezifikationen sowie seine Anforderungen Zuverlässigkeit und Sicherheit betreffend vollständig sind und das System diese erfüllen wird.

Fehlerprävention. Fehlerprävention ist immer Teil beim Entwickeln und Implementieren eines Systems. Beispielsweise ist das Vorbeugen vor Entwicklungsfehlern ein Ziel aller Entwicklungs-Methodologien, sowohl für Software (Information Hiding, Modularisierung etc.) als auch für Hardware.

Fehlertoleranz. Fehlertoleranz zielt auf das Vermeiden von Fehlfunktionen ab; Dies wird versucht durch folgende Techniken zu erreichen [ALR+04]:

- *Fehlverhaltenerkennung* (engl.: error detection): Erkennt das Auftreten eines Fehlers durch
  - simultane Fehlverhalten-Erkennung (engl.: concurrent detection), findet während der normalen Verrichtung des System-Dienstes statt.
  - präventive Fehlverhalten-Erkennung (engl.: preemptive detection), findet statt, während der Systemdienst unterbrochen ist; Das System wird dabei nach latentem Fehlverhalten und Fehlern untersucht.
- *Recovery-Maßnahmen*: Versetzen des Systems von einem Zustand, in dem ihm ein oder mehrere Fehlverhalten und Fehler innewohnen in einen Zustand ohne erkannte Fehlverhalten und Fehler, sodass das System wieder aktiviert werden kann. Dies geschieht mithilfe der Maßnahmen:
  - *Fehlverhalten-Behandlung* (engl.: error handling): Merzt Fehlverhalten aus dem Systemzustand aus, mithilfe der Methoden
    - Rollback: Bringt das System zurück in einen gesicherten Zustand, der gültig war, bevor das Fehlverhalten auftrat. Einen solchen gesicherten Zustand heißt Sicherungspunkt (engl.: checkpoint).
    - Rollforward: Der Zustand ohne erkannte Fehler ist ein neuer Zustand.
    - Kompensation: Im Fehlverhalten behafteten Zustand steht genügend Redundanz bereit, um das Fehlverhalten zu maskieren.
  - *Fehler-Behandlung* (engl.: fault handling): Verhindern, dass Fehler erneut aktiviert werden durch
    - Diagnose: Identifizieren und Aufzeichnen der Art und Ursache eines Fehlverhaltens und die Stelle seines Auftretens.
    - Isolation: Physisches oder logisches Ausschließen der fehlerhaften Komponente vom weiteren Partizipieren beim Verrichten des Dienstes des Systems, das heißt den Fehler in einen latenten Zustand zu versetzen.
    - Rekonfiguration: Entweder das Einsetzen von Ersatzkomponenten oder Aufteilen der Aufgaben unter nicht fehlerhaften Teilen..

- Reinitialisierung: Das Überprüfen, Updaten und Aufzeichnen der neuen Konfiguration und Aktualisieren der Tabellen und Einträge des Systems.

Ein Begriff, der wie das Konzept der Fehlertoleranz Maßnahmen zum Maskieren etwaiger Fehler zusammenfasst, ist *recovery-orientiertes Rechnen* (engl.: recovery-oriented computing, ROC).

Fehlerbeseitigung. Beim Konzept der Fehlerbeseitigung wird selbige während der Systementwicklung und während der Nutzungsphase des Systems unterschieden.

Fehlerbeseitigung in der Entwicklungsphase des Systems. In der Entwicklungsphase des Systems vollzieht sich die Fehlerbeseitigung in drei Schritten: *Verifikation*, *Diagnose*, *Korrektur*. Verifikation ist der Prozess des Überprüfens, ob das System sich an seine vorgegebenen Eigenschaften hält, die auch Verifikationsbedingungen heißen. Ist dies nicht der Fall, werden die beiden anderen Schritte unternommen: Diagnose der Fehler, die verhindern, dass die Verifikationsbedingungen erfüllt werden, anschließend werden die notwendigen Korrekturen vollzogen. Nach der Korrektur sollte der Verifikationsprozess wiederholt werden um sicherzustellen, dass die Fehlerbeseitigung keine unerwünschten Auswirkungen hatte.
Verifikationstechniken werden danach unterschieden, ob ihre Anwendung das Ausführen des Systems erfordert; Ist dies nicht der Fall, so spricht man von *statischer Verifikation*, anderenfalls von *dynamischer Verifikation*.
Statische Verifikation kann durchgeführt werden

- auf dem System selbst, in Form von
  - statischer Analyse, beispielsweise mittels Inspektionen oder Walk-Through, Dataflow-Analyse, Komplexitätsanalyse etc.,
  - Theorem-Beweisen,
- auf einem Modell des System-Verhaltens, wobei das Modell für gewöhnlich ein State-Transition-Modell ist, beispielsweise mit Petrinetzen, endlichen oder unendlichen Zustandsautomaten.

Dynamische Verifikation erfordert das Ausführen des Systems. Dabei wird eine Reihe von Eingaben gemacht, um das Erfülltsein der Verifikationsbedingungen zu überprüfen.

Fehlerbeseitigung während der Nutzungsphase des Systems. In der Nutzungsphase des Systems ist Fehlerbeseitigung entweder eine korrektive Wartung (engl.: corrective maintenance) oder eine präventive Wartung (engl.: preventive maintenance). Erstere zielt darauf ab, Fehler zu beseitigen, die ein oder mehrere Fehlverhalten verursacht haben und gemeldet worden sind. Dagegen strebt die präventive Systemwartung das Aufdecken und Beseitigen der Fehler an, bevor diese während des normalen Betriebs ein Fehlverhalten verursachen können. Wartungsarbeiten können dabei entweder online durchgeführt werden, also ohne Unterbrechung des verrichteten Dienstes, oder offline während eines Dienst-Ausfalls.

Fehlervorhersage. Fehlervorhersage wird vollzogen mithilfe der Auswertung des Systemverhaltens bezüglich des Auftretens und der Aktivierung der Fehler. Diese Auswertung hat zwei Varianten [ALR+04]:

- *Qualitative* oder *ordinale Auswertung*: Sie zielt darauf ab, die verschiedenen Modi der Fehlfunktionen zu identifizieren und zu klassifizieren.
- *Quantitative* oder *wahrscheinlichkeitstheoretische Auswertung*: Sie zielt darauf ab, das Ausmaß, mit dem die einzelnen Attribute erfüllt werden, mit Begriffen der Wahrscheinlichkeit auszudrücken; Solche Attribute heißen dann Maße (engl.: measures). Leistungsbezogene Maße werden unter dem Begriff Leistungsfähigkeit (engl.: performability) zusammengefasst.

So genannte *Zuverlässigkeits-* und *Sicherheitsbenchmarks* sind Anwendungen, mithilfe derer es möglich ist, Maße für das Verhalten eines Computersystems beim Auftreten von Fehlern festzulegen. Sie ermöglichen das Einsetzen verschiedener Techniken der Fehlervorhersage und können auf diese Weise

- die Zuverlässigkeit und Sicherheit eines Systems charakterisieren und
- einen Vergleich zweier alternativer oder konkurrierender Lösungen bezüglich eines oder mehrerer Attribute anstellen.

Abbildung 5 integriert alle betracheten Aspekte von Zuverlässigkeit und Sicherheit in einer Abbildung.

Fazit. Die eben angeführten Konzepte zum Entwerfen und Implementieren zuverlässiger und sicherer Systeme können ihre Ziele niemals vollständig erreichen. Der Grund dafür ist, dass die Entwurfs- und Analysetätigkeiten von Menschen vollzogen werden und eo ipso fehleranfällig sind. Diese Unvollkommenheit zeigt, dass nur der kombinierte Einsatz der Anwendungen aller Konzepte, vorzugsweise bei jedem Schritt der Entwicklung und Implementierung eines Systems, noch am ehesten zur Realisierung eines sicheren, zuverlässigen Computersystem führen kann. Dabei sei angemerkt, dass dieser Prozess der Anwendung der Konzepte zum Erreichen von Zuverlässigkeit und Sicherheit rekursiv ist: Denn gegenwärtige Computersysteme sind zumeist dermaßen komplex, dass zu ihrem Entwurf und ihrer Implementierung der Einsatz diverser Hard- und Software-Tools vonnöten ist, die ihrerseits wieder zuverlässig und sicher sein müssen, etc. [ALR+04].

5 Zuverlässige adaptive Informationssysteme

Im Folgenden werden nun Ziele und Konzepte bei Entwurf und Implementierung von DAIS vorgestellt, auch mit Blick auf deren künftige Entwicklung.

5.1 Entwurfsziele

Gegenwärtige Informationssysteme müssen zu einem bestimmten Grad adaptierbar sein, das heißt, dass diese bei Verminderung der Systemressourcen, beispielsweise der zeitweise Nichtverfügbarkeit einer Datenquelle, bei stark schwankenden Nutzer-

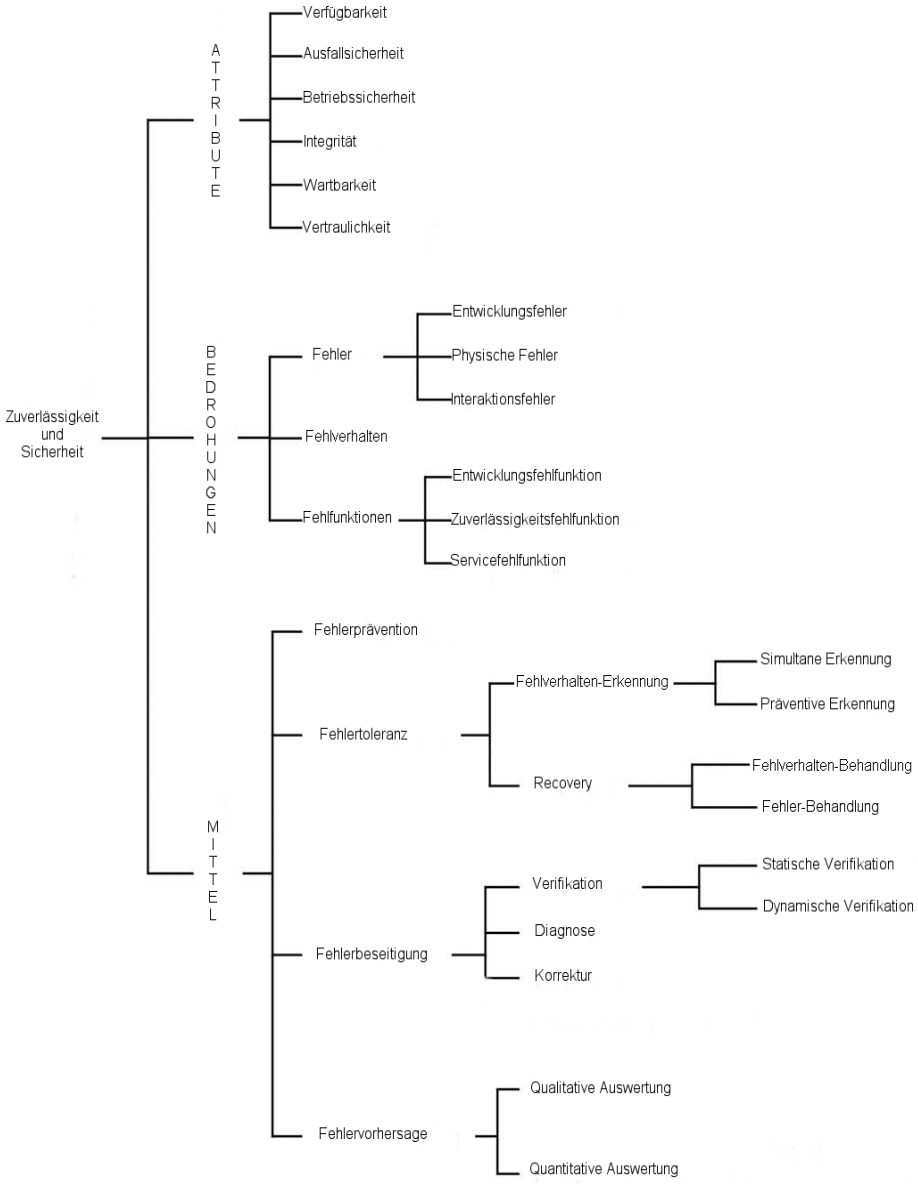

Abbildung 5. Verfeinerter Zuverlässigkeits- und Sicherheitsbaum

zahlen und bei sich häufig ändernden Charakteristika der Operator-Aktionen durch Anpassung bestimmter Systemparameter die von den Nutzern wahrgenommene Leistung des Systems nicht beeinträchtigt wird.

In diesem Zusammenhang ist vor allem für webgestütze IT-Systeme die Skalierbarkeit eine Herausforderung. Um diesen Erfordernissen gerecht zu werden, muss das System zu einem gewissen Grad imstande sein, sich selbst zu überwachen. Entsprechende Konzepte zur Adaptierung werden bei heutigen Informationssystemen jedoch noch wenig eingesetzt.

Verlässliche Informationssysteme zeichnen sich durch den geeigneten Umgang mit dem Auftreten von erwarteten Fehlern aus: Falsche Operator-Aktionen sowie Hardware- und Softwarefehler während des Betriebs eines DAIS können niemals völlig ausgeschlossen werden. Um auf diese Weise drohender Dateninkonsistenz, Datenverlusten oder Nichtverfügbarkeit vorzubeugen, ist das System nach den Prinzipien des recovery-orientierten Rechnens (ROC) zu entwerfen. Die den Informationssystemen zugrunde liegenden Datenbanken haben dieses Konzept bereits erfolgreich umgesetzt, bieten Ansätze zum Handling mit vielen denkbaren Fehlersituationen und können in Fehlerfällen einen transaktionskonsistenten Zustand garantieren; Diese Eigenschaft ist nun bei der Implementierung des gesamten DAIS umzusetzen. Des Weiteren sind Robustheit und ein hohes Maß an Sicherheit weitere Merkmale verlässlicher Systeme.

5.2 Adaptivität in DAIS

Information Hiding und das Aufteilen des Systems in Schichten, die als abstrakte Maschinen fungieren, waren die bisherigen Eckpfeiler beim Entwurf großer erweiterbarer Systeme, besonders der Datenbanksysteme. Das Entwurfsziel einer erhöhten Adaptionsfähigkeit stellt diese probaten Konzepte aber infrage: Denn, wie schon erwähnt, kann das „*Selbst-Bewusstsein*“ (engl.: self-awareness) eines Systems nicht durch bloße Selbst-Beobachtung einzelner Komponenten erreicht werden. Deswegen ist ein gesteigerter Informationsaustausch zwischen den Komponenten vonnöten, um ein holistisches Bild des Gesamtsystems zu erlangen – was zwangsläufig zu neuen Abhängigkeiten führt.

Adaptivität auf Komponentenebene. Bisweilen werden heuristische Mechanismen angewandt, um die richtige statischen Justierung für spezielle Abstimmungen einzelner Komponenten vornehmen zu können, die aber nicht das Potenzial für komplexere Situationen haben. Adaptierung auf Komponentenebene erfordert deshalb das Sammeln statistischer Informationen und muss auf deren Basis fortlaufende Anpassungen seiner Verhaltensparameter oder seiner Leistungsverbesserung vornehmen. Das allgemeine Konzept, um die Selbst*-Eigenschaften eines Systems zu erreichen, besteht aus den vier Phasen Beobachtung, Analyse, Vorplanung und Reaktion, welche die schon erwähnte Onlinefeedback-Kontrollschleife bilden, aus welcher – hoffentlich – die optimalen situationsspezifischen Entscheidungen eruiert werden können.

Adaptivität auf Systemebene. In erster Linie bedeutet Adaptivität auf Systemebene das Bereitstellen flexibler Workflow-Schemata, die sich in unvorhersehbaren Si-

tuationen (neue Kooperationspartner, Nichtverfügbarkeit von Datenquellen etc.) „richtig“ verhalten. Ein wichtiger Aspekt dabei ist das Einhalten bestimmter Zeitschranken bei der Interaktion mit dem Benutzer [Härd05].

5.3 Zuverlässigkeit in DAIS

Zuverlässigkeit erfordert Vorsichtsmaßnahmen für alle erwarteten Fehlfunktionen, was hauptsächlich durch ROC-Mechanismen erreicht wird, also Logging und Recovery, hinsichtlich kritischer Systemanforderungen, beispielsweise hohe Verfügbarkeit. Andere Aspekte wie Sicherheit oder Vertrauen sind aber nichtfunktional und können einer bestimmten Komponente nicht so einfach „dazugefügt“ werden. Um die Zuverlässigkeit eines Systems zu steigern, müssen interne Fehlfunktionen, Fehlfunktionen der Kommunikationskomponenten, Verfälschungen übertragener Daten, Versuche, in das System einzudringen und Anwendungsfehler erkannt werden. Auch dies erfordert, wie die Forderung nach gesteigerter Adaptivität, vom betrachteten System ein gesteigertes Maß an Selbst-Bewusstsein. Um automatisch Abweichungen vom erwarteten Systemverhalten feststellen zu können, benötigt das System Modelle seines Verhaltens und seines umgebenden Umfeldes – beispielsweise in Form einer voreingestellten Richtlinie (engl.: default policy), die das System anweist, wie es in unerwarteten Fällen reagieren soll und durch Zusammenfassen der Bedürfnisse der Nutzer in persönlichen Profilen das System bei unerwarteten Anfragen oder Angriffen unterstützt.

5.4 Zielkonflikt Adaptivität – Zuverlässigkeit

Die eben beschriebenen Mechanismen zum automatischen Anpassen eines Systems beim Administrieren, Erweitern, bei Hard- und Softwareupgrades und bei der Leistungsverbesserung, etc. sollen für ein störungsfreies System bürgen. Schlüssel dazu sind Onlinefeedback-Kontrollschleifen und die Selbst*-Eigenschaften eines Systems, deren Implementierung zwangsläufig die Zahl der Informationskanäle innerhalb des Systems erhöhen wird. Des Weiteren führt die Kooperation zwischen autonomen, viefach heterogenen, Systemkomponenten zu neuen Abhängigkeiten innerhalb des Systems. Diese Tendenzen stehen damit im Widerspruch zu probaten Software-Engineering-Prinzipien wie Modularisierung oder Information Hiding und unterminieren damit die angestrebten Aspekte der Zuverlässigkeit des Systems: Die steigende Komplexität wegen neuer Erweiterungen und verbesserter Adaptivität läuft den für zuverlässige Systeme zwingend gebotenen Vereinfachungen zuwider. Deswegen müssen neue Prinzipien beim Entwickeln zuverlässiger Systeme erarbeitet werden, die allen Aspekten der Zuverlässigkeit gerecht wird. Solche Prinzipien sind möglicherweise

- eine hochgradig modulare Systemstruktur mit
- Komponenten relativ einfacher Funktionalität und
- möglichst einfachen schmalen Komponentenschnittstellen, um die Komplexität der Interaktion zwischen den Komponenten zu minimieren.

Es ist aber unklar, wie das Umsetzen obiger Prinzipien verwirklicht werden kann und wie die übrigen Zielsetzungen hinsichtlich des Systems davon beeinflusst werden, besonders dessen Leistung [Härd05].

5.5 Künftige Entwicklung und Herausforderungen von DAIS

Für ihre Anwendungen müssen künftige DAIS Zugriffe auf viele Datenbanken bieten und viele gleichzeitige Zugriffe koordinieren, Dienste nutzen, deren interne Struktur unbekannt ist und Unterstützung für Geschäftsprozesse bieten, die oft als Workflows entworfen worden sind. Diese DAIS müssen weit mehr noch als heutige mit folgenden Anforderungen fertig werden:

- Weiträumige Verteilung
- Offene Systemstruktur und autonome Komponentenebene
- Heterogenität von Struktur und Inhalt
- Langdauernde interaktive Prozesse

Höhere Offenheit, Verteilung und die Autonomie seiner Komponenten führen dabei zwangsläufig zu weniger Kontrolle über das System, zu neuen Fehlermöglichkeiten und Möglichkeiten böswilliger Angriffe – mit negativen Konsequenzen für die Zuverlässigkeit der Systeme.
Des Weiteren muss ein künftiges DAIS, das beispielsweise Datenquellen aus dem Web benutzt, mit den verschiedenen Arten von Heterogenität und dynamischer Informations-Integration zurechtkommen. Auf Schema-Ebene liegt der Schwerpunkt der Forschung der Informatik auf der XML-Technologie, um diverse Anfrageergebnisse „homogenisieren“ zu können; Auf der Ebene der Daten ist man aber noch weit von einer vollständigen Lösung des Problems entfernt.
Langdauernde Prozesse führen zu komplexen Abhängigkeiten zwischen den Anwendungen und persistenten Datenspeichern, was wiederum das automatische Anpassen an Veränderungen bei der Verfügbarkeit von Ressourcen und die Skalierbarkeit bezüglich der Anforderungen verschiedener Anwendungsszenarien erschwert [Härd05].

6 Zusammenfassung

Informationssysteme sind wichtige Elemente in unserer modernen Gesellschaft, die Abhängigkeit von Wirtschaft und Gesellschaft von Informationssystemen wird tendenziell weiter zunehmen. Dieser Tatsache tragen extreme Anforderungen Rechnung, die an derartige Systeme gestellt werden; Insbesondere sollten sie hochgradig zuverlässig und sicher sein, sowie sich an verschiedenste Situationen automatisch anpassen können. Für die Entwicklung solcher Systeme gibt es bereits eine ganze Reihe von Konzepten und Techniken, es müssen aber noch viele, sowohl technologische wie auch konzeptuelle Hürden genommen werden. Von einer Reihe unmittelbarer Herausforderungen zur Verbesserung gegenwärtiger DAIS dürfte die größte dabei der Zielkonflikt zwischen den beiden zentralen zu verwirklichenden Aspekten Zuverlässigkeit und Flexibilität sein. Aufgrund dieser Problematik, vor al-

lem aber wegen den ganz allgemein dem Menschen und seinen Werken inhärenten Fehlern, dürfte die Verwirklichung der von Jim Gray postulierten ambitionierten „Fernziele“ in voller Allgemeinheit utopisch sein.

Referenzen

[ALR+04] Avizienis, A; Laprie, J; Randell, B; Landwehr, K: Basic Concepts and Taxonomy of Dependable and Secure Computing; Elektronisch verfügbar unter www.loria.fr/~simonot/SlidesCSSEA/IEEETransonDependableComputing2004.pdf

[FedR04] Federal-Reserve-Statistik; Elektronisch verfügbar unter http://www.federalreserve.gov/PaymentSystems/FedWire/fedwirefundstrfann.htm

[GaCo03] Ganek, A; Corbi, G: The dawning of the autonomic computing era; Elektronisch verfügbar unter http://www.www.research.ibm.com/journal/sj/421/ganek.pdf

[Gray99] Gray, J: What's next? A dozen Information-Technology Research Goals; Elektronisch verfügbar unter http://research.microsoft.com/research/pubs/view.aspx?msr_tr_id=MSR-TR-99-50

[Härd05] Härder, T: DBMS Architecture – New Challenges Ahead; Datenbank-Spektrum 2005

[KEF+98] Knight, J; Elder, M; Flinn, J; Marx, P: Summaries of Four Critical Infrastructure Applications; Elektronisch verfügbar unter http://www.dependability.cs.virginia.edu/publications/domainAnalysis.pdf