

Selbstorganisation durch Dezentralität – Dezentralität durch Selbstorganisation: Auf dem Weg zu einem ‚organischen‘ Management von Unternehmens-IT

Stefan Thanheiser, Lei Liu, Hartmut Schmeck

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)
Universität Karlsruhe (TH) im Karlsruhe Institute of Technology (KIT)
Kaiserstraße 12
76128 Karlsruhe
stefan.thanheiser@kit.edu
lei.liu@kit.edu
hartmut.schmeck@kit.edu

Abstract: Die Dezentralisierung von Managementkompetenzen zielt auf eine kollaborative Bewältigung von Entscheidungscomplexität. Eine Voraussetzung hierfür ist, dass die dezentralen Entscheidungsinstanzen ausreichende Fähigkeiten zur Selbstorganisation besitzen. Speziell im IT-Management stieß die Delegation von Managemententscheidungen ‚am Übergang vom Business zur IT‘ aufgrund fehlender Autonomie in Computersystemen bislang an eine Grenze. Die in den letzten Jahren vertiefte Erforschung selbstorganisierender IT-Systeme schafft für die nahe Zukunft Potenzial für die Übernahme von IT-Managementaufgaben durch die IT selbst. Ziel des vorliegenden Artikels ist es, eine konzeptionelle Brücke zwischen ausgewählten Dezentralisierungs- bzw. Selbstorganisations-Konzepten der Business-Seite und der IT-Seite zu schlagen, welche als Ausgangspunkt für weitere Forschung in diesem Schnittstellenbereich dienen kann.

1 Einleitung

Informationstechnologie (IT) ist in Unternehmen zu einem unverzichtbaren Werkzeug für die Bearbeitung unterschiedlichster Problemstellungen geworden. In vielen Branchen ist sie ein (Produktions-)Faktor, welcher besondere Beachtung erfährt: die IT ist für Unternehmen nicht selten erfolgskritisch. Vor diesem Hintergrund wird der betriebliche Einsatz von IT zunehmend aktiv verwaltet: durch Einführung von IT-Governance-, IT-Management- und IT-Controlling-Strukturen versuchen Unternehmen, betriebliche IT strategisch und operativ an Unternehmensziele zu binden. Diese Managementaufgabe gestaltet sich jedoch zunehmend als komplex und ist deswegen – mit steigender Tendenz – personal- und kostenintensiv ([Gil02]). Die zunehmende ‚Industrialisierung‘ betrieblicher IT ([JDZB05]) verringert im Allgemeinen die IT-Fertigungstiefe innerhalb eines Unternehmens. Der ebenfalls wachsende Trend zur Spezialisierung führt jedoch dazu, dass in der Wertschöpfungskette einer IT-Dienstleistung immer mehr Service-Provider

integriert werden müssen ([RSG06, S.16]). Für ein Unternehmen, welches ein IT-Gesamtprodukt entlang seiner Wertschöpfungskette verantwortet, bedeutet dies, dass eine steigende Anzahl von Einzelkomponenten und Schnittstellen – auch im Sinne des IT-Managements und IT-Controllings – orchestriert, überwacht und verwaltet werden muss.

Eine mögliche Strategie gegen steigende Entscheidungskomplexität liegt in der Dezentralisierung von Managemententscheidungen. In der BWL und insbesondere in der Organisationslehre ist Dezentralität ein seit längerer Zeit intensiv diskutiertes Thema. 'Dezentralisierung' impliziert in diesem Kontext die Verlagerung von Entscheidungskompetenz innerhalb von Hierarchien – weg von zentralen Entscheidungsknoten hin zu einzelnen, lokal agierenden Handlungsobjekten.

Eine der Kernvoraussetzungen für diese Art von Dezentralisierung ist jedoch, dass die einzelnen lokalen Handlungsobjekte auch zur Übernahme von Entscheidungskompetenz bzw. zu einer gewissen Selbstorganisation fähig sind. Im sozialen System einer herkömmlichen, aus Menschen bestehenden Managementhierarchie stellt dies durch die Möglichkeit von Schulungen, Teilnehmerauswahl, etc. i.d.R. kein Problem dar. In soziotechnischen Systemen, wie sie beim Management von IT vorliegen, stießen Bemühungen einer Dezentralisierung bislang spätestens beim Übergang von der Business- zur IT-Seite durch fehlende ‚technische Selbstorganisation‘ an ihre Grenzen.

In den letzten Jahren gewinnt ein Forschungszweig in der Informatik an Bedeutung, die sich mit selbst-organisierenden bzw. ‚organischen‘ Computersystemen beschäftigt. Das Positionspapier von GI/ITG/VDE [VDE03] definiert einen „organischen Computer“ als ein selbstorganisierendes System, welches sich den jeweiligen Umgebungsbedingungen dynamisch anpasst. Organische IT-Systeme besitzen die sogenannten ‚self-x‘-Eigenschaften und sind demnach z. B. selbst-konfigurierend, selbst-optimierend, selbstheilend und selbst-schützend. Für den Kontext des IT-Managements schaffen organische IT-Systeme so Potenzial für eine über den bisherigen Stand hinausgehende Möglichkeit einer Dezentralisierung von Entscheidungsverantwortung im IT-Management.

Ziel des vorliegenden Artikels ist es, eine konzeptionelle Brücke zwischen der Business-Seite und der Seite ‚selbstorganisierender IT‘ zu schlagen, welche als Ausgangspunkt für weitere Forschung in diesem Schnittstellenbereich dienen kann. Besonders von Interesse für diesen Artikel wird es zudem sein, bestehende dezentrale Steuerungskonzepte der Business-Seite mit entsprechenden Konzepten der IT-Seite abzugleichen und ggf. in Einklang zu bringen.

Der Artikel ist im weiteren Verlauf folgendermaßen gegliedert: zur Beschreibung der ‚Business-Seite‘ werden in Abschnitt 2 Managementkonzepte charakterisiert, welche dezentrale Selbstorganisation als Paradigma nutzen. In Abschnitt 3 wird der aktuelle Forschungsstand der ‚IT-Seite‘ im Bezug auf selbstorganisierende IT-Systeme vorgestellt. Abschnitt 4 versucht, einen Abgleich der Konzepte vorzunehmen sowie die Vision eines ‚organischen‘ IT-Managements zu entwerfen, welches soziale Managementhierarchien der Business-Seite mit technischen Managementhierarchien einer nach Konzepten des Organic Computing gestalteten Unternehmens-IT integriert. Der letzte Abschnitt des

Artikels fasst die Inhalte zusammen und gibt einen Ausblick auf weitere Forschungsthemen im Kontext.

2 Selbstorganisation durch Dezentralität auf der Business-Seite

Ein traditionelles Verständnis von (IT-)Management definiert ein zentrales Planungs-, Kontroll-, Informations- und Koordinationssystem für das Unternehmen und dessen IT. Doch wachsende Komplexität und gestiegene Dynamik der Unternehmens-(Um-)welt machen eine zentralisierte Detailsteuerung betrieblicher IT zunehmend schwierig. In diesem Kontext zielen die in den letzten Jahren entwickelten *dezentralen* Planungs-, Kontroll-, Informations- und Koordinationsstrukturen i.d.R. darauf ab, zentralistische ‚Command and Control‘ zugunsten von Netzwerken dezentralisiert und autonom entscheidender – jedoch über relative Zielvorgaben und gemeinsame Ziele dirigierter – Entscheidungsträger abzuschaffen. Dezentrale Managementkonzepte fördern und fordern somit das Entstehen lokaler Selbstorganisation innerhalb der Unternehmenshierarchie. Als Beispiel für diese Vorgehensweise ist insbesondere das Managementkonzept des ‚Beyond Budgeting‘ (vgl. [HF03]) zu nennen.

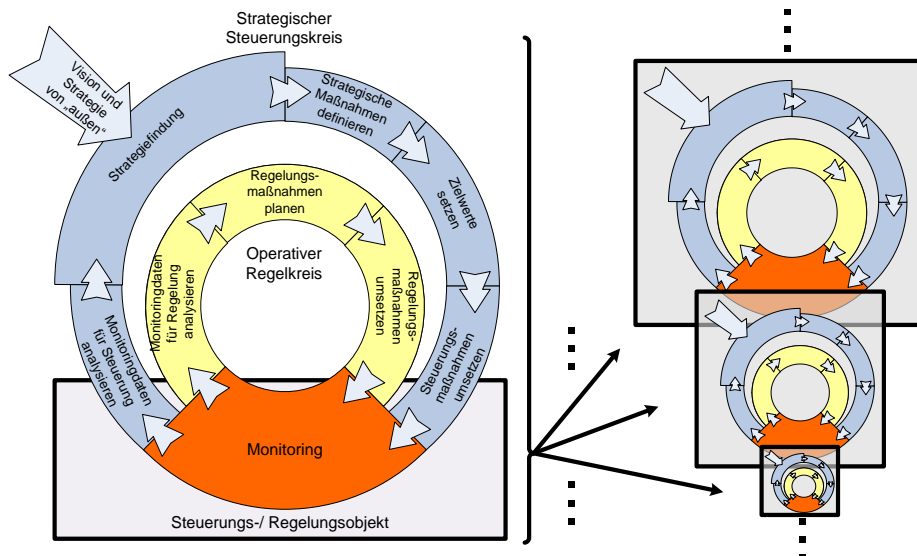


Abbildung 1 - Idealtypischer Steuerungs-/Regelkreis und dessen Vermaschung in einer Hierarchie

Die verbleibenden zentralen Strukturen sind bei Dezentralisierung einem Wandel unterworfen: von Schaltzentralen einer detaillierten Ablaufsteuerung hin zu Instrumentarien zur Anleitung und Führung der dezentralen Einheiten. ‚Führung‘ besteht in diesem Kontext darin, Spielräume und Grenzen der dezentralen Selbstorganisation zu definieren – und auch dezentrale Zielerreichung aus globaler Sicht zu messen und zu beurteilen. Insgesamt ergibt sich durch diese Maßgabe ‚gesteuerter Selbstorganisation‘ ein Gesamtsystem aus zentralen und dezentralen Entscheidungsträgern, welche die Managementaufgaben ‚kollaborativ‘ wahrnehmen.

Management als Prozess wird oft mit einem Steuerungs- bzw. Regelkreislauf in Verbindung gebracht (Abbildung 1). Bei der Delegation von Management entsteht innerhalb eines Unternehmens aus wenigen zentralen Regelkreisen globalen Bezugs eine Vielzahl neuer dezentraler Regelkreise mit lokalem Bezug. Zur Realisierung von *gesteuerter* Selbstorganisation ist eine umfassende Koordination bzw. ‚Synchronisation‘ aller Regelkreise notwendig: eine übergeordnete Regelkreisinstanz betreibt hierbei ‚Meta-Management‘ einer oder mehrerer anderer Regelkreis-Instanzen. Hahn und Hungenberg sprechen im Kontext von einem vermaschten Führungsprozess ([HH01, S.50-55]).

Ein häufig im Kontext gesteuerter Selbstorganisation im Unternehmen diskutiertes ([Ja00], [Ma06], u.a.) Konzept ist das von Stafford Beer formulierte ([Be59]) ‚Viable Systems Model‘ (VSM) – im deutschsprachigen Raum als ‚Modell lebensfähiger Systeme‘ bekannt. Das VSM erhebt den Anspruch, Strukturen und Abhängigkeiten zu definieren, die bei jedem selbstorganisierenden bzw. ‚lebensfähigen‘¹ System - unabhängig von dessen Ausprägung oder Hierarchiestufe – in analoger Form vorzufinden seien. Unabhängig von diesem Anspruch soll im vorliegenden Artikel das VSM als besondere Systematik kollaborativ arbeitender Steuerungs-/Regelungskreise vorgestellt werden, welche speziell auf unterschiedliche Aspekte von Selbstorganisation eingeht. Im VSM besitzt jedes einzelne lebensfähige System (LS) zwingend je eine Instanz aus einer Gruppe von fünf Untersystemtypen. Jeder dieser Untersystemtypen besitzt einen festen Umfang an Aufgaben und Funktionen und enthält für sich selbst je einen eigenen Management-Regelkreis.

Zu den Untersystemtypen des VSM im Einzelnen (Abbildung 2): Während mehrere Instanzen des ‚Systems 1‘ unterschiedliche Handlungsfelder des Gesamt-LS sowie die auf die einzelnen Handlungsfelder fokussierenden Regelkreise repräsentieren; verkörpert das ‚System 2‘ eine Controlling-Funktion, welche die operative Tätigkeit in den einzelnen Handlungsfeldern koordiniert und nach allgemeinen Maßgaben überwacht. Das ‚System 3‘ ist im VSM die für alle Handlungsfelder des LS verantwortliche Managementinstanz, welche den kurzfristigen, internen Operationsplan des Gesamtsystems erstellt. Es fokussiert auf die Herstellung eines internen Gleichgewichtes zwischen den einzelnen Handlungsfeldern. Im Kontrast hierzu steht das ‚System 4‘, welches durch ganzheitliche Umweltbeobachtung die ‚Augen und Ohren‘ des LS stellt. Durch Fortschreibung der von intern und extern bei sich eintreffenden Informationen stellt das ‚System 4‘ zudem eine eher auf den strategischen Kontext gerichtete Planungsfunktionalität zur Verfügung. Das ‚System 5‘ stellt schließlich das ‚normative Management‘ des LS dar – im ihm manifestieren sich die grundlegenden Normen und Regeln für das Gesamtsystem. In ihm findet auf Ebene globaler Systemolicies ein Abgleich von Soll- und Ist-Systemzuständen statt.

Jedes Teilsystem steht mit den anderen Untersystemtypen in sog. ‚homöostatischen Lenkungsbeziehungen‘. Das von Ashby beschriebene Prinzip der Homöostase ([As57]) referenziert im Kontext des VSM auf eine systemautonome Regulation interner System-

¹ Der Begriff der Lebensfähigkeit schließt hierbei u.a. die Fähigkeiten zur Selbstorganisation und der Lenkbarkeit mit ein. Der Term ‚Lebensfähigkeit‘ bezieht sich nicht die auf Lebensfähigkeit im biologischen Sinn. Er bezieht sich auf die Eigenschaften eines Systems bezüglich Lenkungs-fähigkeit, Kommunikationsfähigkeit, Anpassungsfähigkeit, Lernfähigkeit, Entwicklungsfähigkeit, Gestaltbarkeit und Selbstorganisationskraft.

zustände mit dem Ziel, trotz systeminterner oder -externer Störungen einen überlebenswichtigen Gleichgewichtszustand beizubehalten. Die Regelkreise der einzelnen Untersysteme sind somit in einer Hierarchie ‚vermascht‘: sobald eine Störung bei einem der Untersysteme eines LS auftritt, sorgt das Zusammenwirken aller Steuerungskreisläufe der anderen VSM-Untersysteme dafür, dass sich das LS insgesamt auf die neuen Gegebenheiten einstellt.

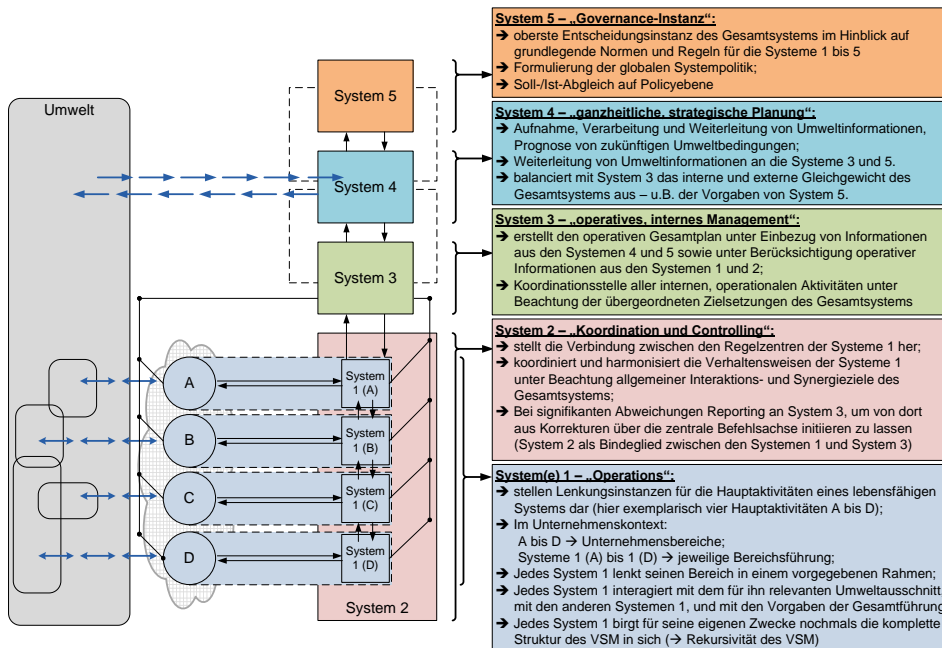


Abbildung 2 - Das Viable Systems Model (VSM) nach Stafford Beer
 (eigene Darstellung in Anlehnung an [Ma06 S.86ff.])

Das für das VSM essenzielle ‚Prinzip der Rekursivität‘ besagt, dass jedes LS ein Subsystem mindestens eines anderen LS ist – und selbst wiederum aus mindestens einem LS besteht. Im Kontext eines dezentral organisierten Unternehmens gesprochen: Ein Bereich ist Teil eines Unternehmen, besitzt für seinen Bereich jedoch in gewissem Umfang Handlungsfreiheit. Bereiche bestehen weiterhin aus weitgehend für sich autonomen Abteilungen; jede Abteilung besteht wiederum aus einer Anzahl von Mitarbeitern, welche ebenfalls für ihren konkreten Arbeitsbereich gewisse Handlungsfreiheit besitzen. Die Koordination dieser verschiedenen Kompetenzbereiche behandelt das VSM im sog. ‚Prinzip der Autonomie‘: jedes LS genießt zwar prinzipiell völlige Verhaltensfreiheit – besitzt als Teil eines übergeordneten LS jedoch auch Verantwortung für dessen Weiterbestand. Durch diese Verantwortung wird die prinzipielle Handlungsfreiheit wiederum eingeschränkt; die Selbstorganisationsfähigkeit eines LS wird ‚kanalisiert‘, wobei ein übergeordnetes LS Meta-Management für untergeordnete LS betreibt.

Insgesamt spannt das VSM in ‚top-down‘-Richtung ein Netzwerk autonomer Entscheidungsinstanzen auf. Für den Kontext des vorliegenden Artikels ist interessant, inwiefern

sich dieses Netzwerk ‚nach unten‘ in die betriebliche IT erweitern lässt. Voraussetzung hierfür ist das Vorliegen maschineller Selbstorganisation, welche eine Dezentralisierung von Entscheidungsbefugnissen in den technischen Kontext hinein erst möglich macht. Der folgende Abschnitt beschäftigt sich mit bestehenden Selbstorganisationskonzepten der IT-Seite.

3 Dezentralität durch Selbstorganisation - Konzepte der IT-Seite

Im Kontext *selbstorganisierender* IT-Systeme haben seit 2002 verschiedene Forschungseinrichtungen und Firmen Initiativen gestartet – so z.B. das Forschungsprogramm zu ‚Organic Computing‘ (OC) ([OC07]) oder das ‚Autonomic-Computing‘-Programm (AC) ([AC07]). Während – mit Bezug auf betriebliche IT – beim Autonomic Computing Standardaufgaben im Systemmanagement durch geeignete Rückkopplungsmechanismen selbsttätig erledigt werden sollen, fokussiert sich das Organic Computing auf vernetzte adaptive Systeme, die i.d.R. zweckneutral über eine ausreichende Freiheit zur autonomen Reaktion verfügen sollen, hierbei aber weiter für den Menschen steuerbar bleiben müssen. Die Gesellschaft für Informatik definiert in diesem Zusammenhang einen ‚organischen Computer‘ als ein selbstorganisierendes System, welches sich den jeweiligen Umgebungsbedürfnissen dynamisch anpasst ([VDE03]). Organische IT-Systeme besitzen sog. ‚self-x‘-Eigenschaften und sind demnach z. B. selbstkonfigurierend, selbstoptimierend, selbstschützend, selbstheilend und selbsterklärend, ermöglichen jedoch gleichzeitig eine ‚gesteuerte Selbstorganisation‘ (siehe z. B. [Sc05]).

Essenziell für die Realisierung des erwünschten Grads an Selbstorganisation ist eine innerhalb eines Systems zu etablierende Steuerungseinheit – im Fall des OC als ‚Observer-/Controller-Einheit‘ (O/C) bezeichnet. Nach Richter et al. nimmt der ‚Observer‘ ein beobachtetes System und dessen Umwelt per sensorischem Input wahr und gibt diese Daten an den ‚Controller‘ weiter, der anhand dieser Informationen auf verschiedene Arten auf das System Einfluss nehmen kann ([Ric06]).

Eine O/C-Einheit (siehe Abbildung 3) überwacht bzw. steuert ein einzelnes System bzw. einen Systemteil. Hierbei kann diese entweder systemextern auf das ‚System under Observation and Control‘ (SuOC) zugreifen oder die O/C-Einheit ist systemintern innerhalb des SuOC untergebracht [Ric06] [Cak07]. SuOC und die dazugehörige O/C-Einheit bilden die Grundarchitektur eines ‚organischen Systems‘. Die grundsätzliche Arbeitsweise einer O/C-Einheit lässt sich mit dem aus dem AC stammenden Akronym ‚MAPE‘ beschreiben – nämlich ‚Monitor‘, ‚Analyze‘, ‚Plan‘ und ‚Execute‘ (vgl. [KC03]).

Mit ‚Monitor‘ werden vom SuOC Daten in Form von Kennzahlen und/oder Konfigurationsdaten erhoben, aggregiert, gefiltert und weitergemeldet. Unter ‚Analyze‘ sind alle Mechanismen zusammengefasst, die eine komplexe Modellbildung möglich machen (z.B. Verfahren für Korrelation und Zeitreihenanalyse). ‚Plan‘ setzt die bisher erfolgte Datenanalyse unter Berücksichtigung vorgegebener Zielangaben und Randbedingungen in konkrete Ziele und Steuerungsmaßnahmen um. ‚Execute‘ sorgt schließlich für die tatsächliche Umsetzung der Steuerungsmaßnahmen. Ein wesentlicher Aspekt der O/C-Architektur ist die Fähigkeit zur selbsttätigen Adaption – einerseits realisiert über online-

und offline-Lernmechanismen im Controller, andererseits durch die Möglichkeit der situationsangepassten Veränderung des Beobachtungsmodells im Observer. Für eine detailliertere Darstellung des Innenlebens einer O/C-Struktur verweisen wir an dieser Stelle auf Abbildung 3 sowie den Artikel [Ric06], welche die Aufgaben in MAPE in einzelne Module und Modulaufgaben innerhalb der O/C-Einheit detaillieren.

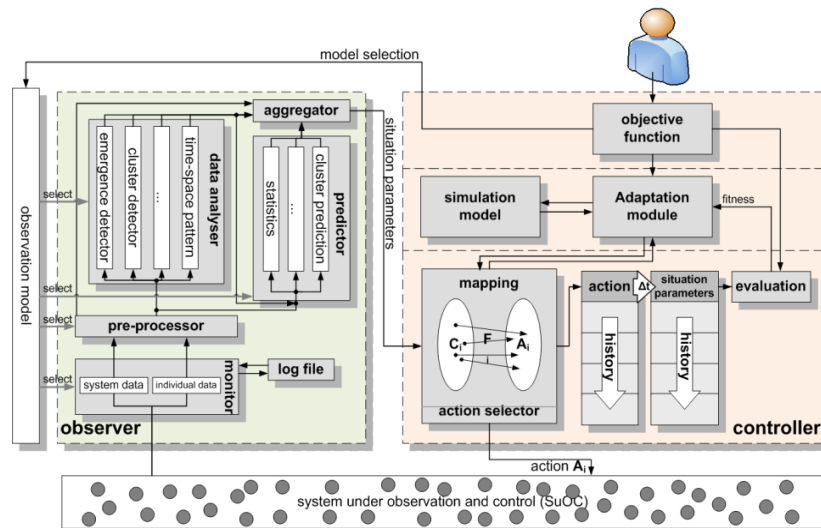


Abbildung 3 - Detailstruktur einer Observer/Controller-Einheit
(eigene Darstellung in Anlehnung an [Bra06])

Innerhalb einer IT-Landschaft im betrieblichen Umfeld wurden in der Literatur bereits mehrere Einsatzszenarien von O/C-Einheiten diskutiert: neben selbstorganisiertem Incident- und Problem-Management auf Infrastrukturebene ([Bri07]) bestehen erste Konzepte für die Einführung mehrdimensionaler Selbstorganisationshierarchien auf allen Ebenen einer Service-orientierten Architektur ([LTS07] und [TLS07]).

Zusammenfassend lässt sich feststellen, dass eine O/C-Einheit für das SuOC die einzelnen Teilaufgaben des in Abschnitt 2 beschriebenen Management-Regelkreises umsetzt. Durch die autonome Abwicklung des Management-Regelkreises stellt ein organisches System in einem bestimmten Kontext Selbstorganisation her. Diese lokale Selbstorganisationsfähigkeit erlaubt es, in einem Systemverbund (wie z. B. einer SOA) Steuerungsverantwortung dezentral zu (re-)organisieren – Dezentralität wird durch anpassungsfähige Selbstorganisation ermöglicht.

4 Auf dem Weg zu einem ‚organischen‘ IT-Management

Der folgende Abschnitt versucht, die Vision eines ‚organischen‘ IT-Managements zu entwerfen, welches soziale Managementhierarchien der Business-Seite mit technischen Managementhierarchien einer nach Konzepten des Organic Computing gestalteten Unternehmens-IT integriert. Zu diesem Zweck ist es notwendig, die in den vorhergehenden

Abschnitten charakterisierten Konzepte abzugleichen und miteinander zu verknüpfen. Darüber hinaus soll - anhand eines Beispiels aus dem IT Service Management - das soziotechnische Zusammenspiel in einem organischen IT-Management modellhaft illustriert werden.

4.1 Ist ein organischer Computer ‚lebensfähig‘?

Um die Frage nach der ‚Lebensfähigkeit‘ eines organischen Systems im Sinne des VSM zu beantworten, sollen die Konzepte im OC mit den zentralen Elementen des VSM abgeglichen werden:

- (a) *Handlungsfelder (Systeme 1)*: Auch ohne explizite Aussagen darüber, ob eine O/C-Einheit ein SuOC im Bezug auf mehrere unterschiedliche Kriterien beobachtet und steuert, ist es aufgrund des generischen Ansatzes klar, dass sich „objectives“ und „Evaluationskriterien“ auch auf mehrere – potentiell gegenläufige – Kriterien beziehen können. Im einfachsten Fall sieht eine O/C-Einheit lediglich ein spezielles Handlungsfeld für die Beobachtung/Steuerung seines SuOC vor – die generische Architektur wurde jedoch für komplexere Anwendungsszenarien konzipiert, um ein komplex strukturiertes SuOC und dessen Umwelt auf mehreren Ebenen auch ‚multikriteriell‘ verwalten zu können.
- (b) *Koordination und Führung von Handlungsfeldern (System 2 / System 3)*: Die OC-Literatur trifft keine expliziten Aussagen darüber, wie unterschiedliche Handlungsfelder eines organischen Systems koordiniert und ausgeglichen werden können, dies ist auch nur bezüglich spezieller Anwendungsfelder genauer spezialisiert werden. Die Konzeption des Controllers schließt jedoch über das Adaptionmodul und die anpassungsfähige Zuordnung von Aktionen zu vorliegenden Situationen die Fähigkeit zu multikriterieller Optimierung und somit zur Koordination mehrerer Handlungsfelder ein.
- (c) *Schaffung eines internen und externen Gleichgewichtszustandes (System 4)*: Çakar et al. diskutieren in [Ric06, Cak07] Möglichkeiten der Interaktion zwischen O/C-Einheit und SuOC. Wesentliches Ziel ist, dass das SuOC ohne externe Beeinflussung seine Aufgaben erfüllt, d.h. dass der Controller nur dann eingreift, wenn Zielvorgaben nicht eingehalten werden oder eine unerwünschte Abweichung vorausgesehen wird. Eine situationsgetriggerte Anpassung des Observers sowie die Lernfähigkeit des Controllers ermöglichen eine situationsgerechte Steuerung des SuOC. Insofern ist die Schaffung eines internen und externen Gleichgewichtszustandes auch ein wesentliches Ziel des Organic Computing.
- (d) *Bindung der Selbstorganisation an Normen und Regeln (System 5)*: Ein organischer Computer soll per Definition vom Menschen kontrollierbar sein und muss sich an dessen Zielvorgaben orientieren. Dies ist ein wesentlicher Bestandteil des Konzepts der „gesteuerten Selbstorganisation“, in der die Güte des beobachteten Verhaltens stets anhand der von außen vorgegebenen Randbedingungen gemessen wird. Die vorgegebenen Zielvorgaben und Randbedingungen bilden sogenannte ‚Policies‘. Policies können z. B. vom Menschen vorgegeben werden oder sie entstehen durch Verpflichtungen gegenüber anderen organischen Systemen (z. B. über Vereinbarung von sog. ‚Service Level Agreements‘) [KC03].

- (e) *Horizontale und vertikale Kollaboration / Steuerung der Selbstorganisation:* Zur Realisation technischer Hierarchien von organischen Einzelsystemen werden in der OC-Literatur vermaschte O/C-Strukturen vorgeschlagen, welche eine zentrale Rolle bei der Erzeugung ‚erwünschter‘ Eigenschaften des Gesamtsystems spielen [SMS05]. Jedes der organischen Einzelsysteme bestimmt sein internes Verhalten und seine Beziehung zu anderen Einzelsystemen grundsätzlich autonom. Die Einzelsysteme bilden eine Art Hierarchie bzw. ein Verantwortungsgefüge: Übergeordnete Einzelsysteme erhalten über Policies Zielvorgaben, für deren Erfüllung sie alleine verantwortlich sind. Sofern eine gewünschte Leistung nicht vom verantwortlichen Einzelsystem selbst erbracht werden kann, so muss dieses die Leistung per ‚subcontract‘ [KC03] bei anderen Einzelsystemen beauftragen.

Insgesamt zeigt sich, dass in den Konzepten des Organic Computing die Kernansätze eines ‚lebensfähigen Systems‘ im Sinne des ‚Viable Systems Model‘ grundsätzlich vorhanden sind. Bei geeigneter Implementierung und Einbettung in eine VSM-Umgebung kann ein organisches IT-System innerhalb seines Verantwortungsbereiches als ‚lebensfähig‘ im Sinne des VSM bezeichnet werden.

4.2 Organisches IT-Management – im IT Service Management

Im Abschnitt 3 war im Kontext des Organic Computing von durch Selbstorganisation ermöglichter Dezentralität die Rede. In der Gegenrichtung (‚Selbstorganisation durch Dezentralisierung‘) wird es für das IT-Management möglich, ‚lebensfähige‘ organische IT-Systeme gezielt für die Übernahme von bisher vom Menschen bewältigter IT-Komplexität einzusetzen. Dies könnte durch gezielte Schaffung autonomer maschineller Instanzen geschehen, an welche einfache (IT-)Managementtätigkeiten delegiert werden können. Ein ‚organisches‘ IT-Management erweitert in diesem Kontext die Ansätze des Organic Computing für die betriebliche IT durch systematische Integration der Business-Seite.

Nach dem Rekursivitätsprinzip – welches auch für ‚lebensfähige‘ organische Systeme gilt – ist ein durchgängiges Delegationsgefüge denkbar – vom obersten IT-Management eines Unternehmens bis auf die Ebene eines einzelnen organischen Computersystems. Abbildung 4 zeigt dies am Beispiel einer Delegationskette, welche sich top-down von der Ebene der IT Governance über die soziotechnischen Systeme des IT Managements, des IT Service Management, der verschiedenen ITIL-Disziplinen bis in eine ‚organische Service-orientierte Architektur‘ ([LTS07], [TLS07]) hinein erstreckt.

Besonderes Kennzeichen eines organischen IT Managements ist, dass die Managementaufgabe innerhalb der Hierarchie durch ‚Mensch und Maschine‘ kollaborativ wahrgenommen wird. Den menschlichen wie maschinellen Akteuren auf jeder Ebene wird gestattet, sich innerhalb ihres spezifischen Kontextes selbst zu organisieren. Jeder so entstandene autonome Regelkreis ist in einer Hierarchie vermascht und lässt sich auf diese Weise einen ‚Autonomierahmen‘ bzw. eine Policy zuweisen. Die auf einer Rekursions-ebene getroffenen Zieldefinitionen werden zu Policies der darunterliegenden Rekursions-ebenen. In einer herkömmlichen, rein ‚menschlichen‘ Managementhierarchie stellt diese Vorgehensweise keine Neuerung dar. Der Übergang zur IT-Seite gestaltet sich bei

einer vollständig organischen IT jedoch insofern neu, als für das menschliche IT-Management im Regelfall keine Detailsteuerung der IT mehr nötig sein wird: die betriebliche IT organisiert sich im Rahmen der vom Menschen vorgegebenen Grenzen selbst; sie sorgt autonom für die Einhaltung der vom Menschen vorgegebenen Leistungsziele und sorgt darüber hinaus selbsttätig für die Informationsversorgung eines menschlichen IT-Managements. Die organische Unternehmens-IT kümmert sich selbsttätig um den ‚Betrieb nach menschlicher Vorgabe‘, das menschliche Management kann sich auf Definition und Vorgabe von Zielen konzentrieren. Eine Eskalation zum menschlichen IT-Management würde nur dann nötig sein, wenn sich die organische IT nicht mehr ‚selbst helfen kann‘. Bei einer vollständig organischen Unternehmens-IT (Abbildung 4) erfolgt der ‚Managementwechsel von Mensch auf Maschine‘ auf der Prozessebene: Zielvorgaben werden bspw. auf Basis von Wunsch-Qualitätsparametern einzelner Dienstleistungstransaktionen vorgegeben.

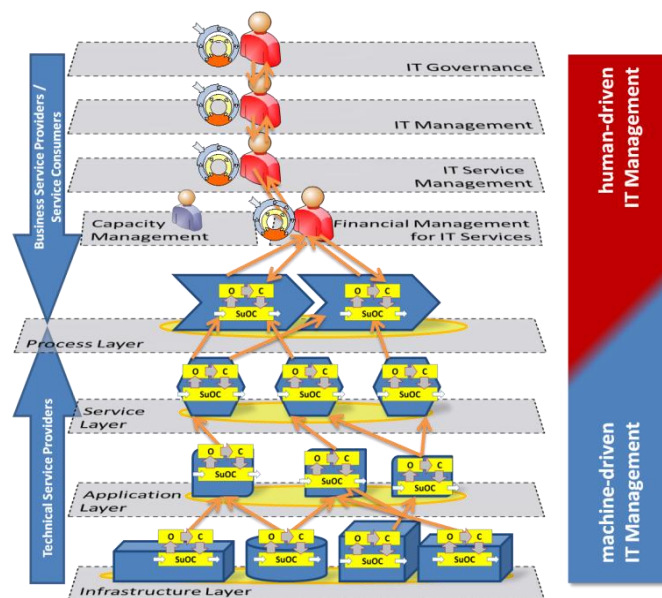


Abbildung 4 – Delegationsgefüge in einem ‚organischen‘ IT-Management

Insbesondere im Kontext des IT Service Managements (ITSM) existiert eine Vielzahl von Anwendungsfällen, bei welchen autonome Systeme durch erweiterte Delegationsmöglichkeit potenziell Nutzen stiften. So könnte insbesondere der ITIL-Service-Delivery-Bereich vom Einsatz organischer IT-Systeme profitieren – insbesondere weil zwischen den dort angesiedelten ITIL-Disziplinen oft ein ‚multikriterieller Ausgleich‘ zwischen bisweilen konkurrierenden Zielen vonnöten ist.

Die Delegation von Managemententscheidungen an selbstorganisierende IT soll am Beispiel der ITIL-Disziplin ‚Financial Management for IT-Services‘ (FM) erläutert werden. Das FM ist für die Verwaltung, Planung und Erlösdeckung der Kosten einer IT-Dienstleistung verantwortlich. Wie aus Abbildung 4 ersichtlich, ist das ‚menschenzentrierte‘ FM in seiner autonomen Tätigkeit der normativen Regulierung durch die Corpo-

rate Governance, die IT Governance und das generelle IT-Management unterworfen. Mit einer top-down Vorgabe zur Kostensenkung bzw. Effizienzsteigerung bzgl. der betrieblichen IT wird in allen Fällen das FM aufgerufen sein, seinen Beitrag zur Erreichung der Ziele der übergeordneten Managementinstanzen zu leisten.

Aus der Fähigkeit organischer IT-Systeme, sich innerhalb von Rahmenvorgaben selbst zu organisieren, entstehen für das FM Vorteile: So wird es ihm bspw. ermöglicht der organischen IT Kostenziele vorzugeben. Die Maßgabe, bspw. für einen Durchlauf eines Dienstleistungsprozesses unterhalb eines bestimmten Kostenlimits zu bleiben (z. B. „Nicht mehr als X Euro pro individuellem End-to-End-Prozessdurchlauf“) wird zu einer der Rahmenbedingungen der organischen SOA-Elemente. Auf diese Weise wird beispielsweise ein organischer SOA-Service ermächtigt, innerhalb des Kostenrahmens die für seine Erstellung notwendigen Applikationen bzw. Infrastrukturelemente u.a. nach Kostengesichtspunkten auszuwählen. Gleichzeitig wird der organische SOA-Service jedoch auch gezwungen sein, unterschiedliche Zielsetzungen ‚auszubalancieren‘: wenn mehrere zuarbeitende Elemente dem Service den geforderten Mindest-Servicelevel (Verfügbarkeiten, Antwortzeiten, etc. – Vorgaben bspw. aus dem Service Level Management) garantieren, wird der organische Service aus diesem Kreis das kostengünstigste Element auswählen, um auch die Zielvorgaben aus dem Financial Management zu erfüllen.

5 Zusammenfassung und Ausblick

Im vorliegenden Artikel wurden sowohl für die Business-Seite als auch für die IT-Seite ausgewählte Konzepte für Dezentralität und Selbstorganisation vorgestellt. Es konnte - im Hinblick auf die Vision eines ‚organischen‘ IT-Managements - eine konzeptionelle Brücke zwischen dem Viable Systems Model als Paradigma der Business-Seite und den Konzepten des Organic Computing hergestellt werden. Ein Vergleich der Kernelemente beider Seiten führte zu dem Schluss, dass organische IT-Systeme bei geeigneter Gestaltung als lebensfähige Systeme im Sinne des VSM gelten können und auch deshalb in eine isomorph vermaschte Managementhierarchie integriert werden können. Nutzen einer solchen Hierarchie wäre, dass sie in top-down-Richtung gesteuerte Selbstorganisation ermöglicht und bottom-up die Einhaltung der Rahmenvorgaben der jeweils übergeordneten Managementschicht anstrebt.

Insgesamt zeigt sich, dass IT-Selbstorganisation das Potenzial besitzt, den top-down bisher an der Grenze zur betrieblichen IT gestoppten Delegationsbestrebungen des IT-Managements neue Möglichkeiten eines ‚organischen‘ IT-Managements zu eröffnen. Es zeigt sich auch, dass in der BWL bereits seit längerer Zeit vorhandene Konzepte ‚gesteuerter, menschlicher Selbstorganisation‘ richtungsweisend für die Informatik sein können – so lässt sich aus dem VSM z. B. die Aufgabe ableiten, sich im Organic Computing verstärkt mit der autonomen Koordination interdisziplinär verknüpfter Akteure zu beschäftigen. Darüber hinaus wird es – neben anderen Themen – von Interesse sein:

- wie ein organisches System in die Lage versetzt werden kann, Fehlentwicklungen rechtzeitig zu erkennen und ggf. proaktiv gegenzusteuern;
- mit welchen Mechanismen ein „Sozialsystem“ aus organischen Systemen ge-

schaffen werden kann – welches möglichst ohne menschlichen Eingriff positive Leistungsbeiträge belohnt; Fehlverhalten jedoch sanktioniert;

Literaturverzeichnis

- [AC07] Internetseite zum 'Autonomic Computing': <http://www.research.ibm.com/autonomic> - letzter Abruf am 03.12.2007
- [As57] Ashby, W. R.: An Introduction to Cybernetics (2. Ed), Chapman & Hall London, 1957
- [Bra06] Branke, J.; et al.: Organic Computing - Addressing Complexity by Controlled Self-organization. In: Margaria, Tiziana; Philippou, Anna; Steffen, Bernhard: Proceedings of ISoLA 2006. Paphos, Cyprus. (IEEE-ISoLA), November 2006, S. 200–206
- [Be59] Beer, S.: Cybernetics and Management. The English Universities Press, 1959.
- [Cak07] Çakar, E., Mnif, M., Müller-Schloer, C., Richter, U., Schmeck, H.; Towards a quantitative notion of self-organisation. In: Proceedings of the 2007 IEEE Congress Evolutionary Computation (CEC 2007), Singapore, 2007, S. 4222–4229.
- [Gil02] Gillett, F. E. et al; Forrester Research Report April 2002 - Organic IT. April 2002
- [HF03] Hope, J.; Fraser, R.: Who needs Budgets? In: Harvard Business Review, Vol. 81, No. 2, Februar 2003, S. 108–116
- [HH01] Hahn, D.; Hungenberg, Harald: PuK - Planung und Kontrolle, Planungs- und Kontrollsysteme, Planungs- und Kontrollrechnung. 6.Aufl., Gabler, 2001.
- [Ja00] Jackson, M. C.: Systems approaches to management. Kluwer Academic/Plenum, New York, USA, 2000.
- [JDZB05] von Jouanne-Diedrich, H.; Zarnekow, R.; Brenner, W.: Industrialisierung des IT-Sourcings. In: HMD - Praxis der Wirtschaftsinformatik 245, 2005, S. 18–27
- [KC03] Kephart, J. O.; Chess, D. M.: The Vision of Autonomic Computing. In: IEEE Computer, Vol.36, No.1, Januar 2003, S. 41–50
- [LTS07] Liu, L.; Thanheiser, S.; Schmeck, H.: Coping with the Complexity of Service-oriented Computing using Controlled Self-Organization. In Di Penta, M.; Melideo, M. (Hrsg.), Proceedings of the SOC@Inside'07 - Service Oriented Computing: a look at the Inside, September 17, 2007; Vienna, Austria. EU Commission, September 2007.
- [Ma06] Malik, F.: Strategie des Managements komplexer Systeme - Ein Beitrag zur Management-Kybernetik evolutionärer Systeme. 9.Auflage, Haupt-Verlag Bern, 2006.
- [OC07] Internetseite zum Thema 'Organic Computing': <http://www.organic-computing.de> - letzter Abruf am 03.12.2007
- [Ric06] Richter, U.; et al.: Towards a generic observer/controller architecture for Organic Computing. In: Hochberger, Christian; Liskowsky, Rüdiger (Hrsg.): INFORMATIK 2006 - Informatik für Menschen!, Bonner Köllen Verlag, 2006, S. 112–119
- [RSG06] Rüter, A.; Schröder, J.; Goldner, A.: IT-Governance in der Praxis. Springer, 2006
- [Sc05] Schmeck, H.: Organic Computing. Künstliche Intelligenz (3/05). July 2005, S. 68-69.
- [SMS05] Schöler, T. and C. Müller-Schloer. An Observer/Controller Architecture for Adaptive Reconfigurable Stacks. In: Proceedings of the 18th International Conference on Architecture of Computing Systems (ARCS), Springer, 2005, S. 139-153.
- [TLS07] Thanheiser, S.; Liu, L.; Schmeck, H.: Towards Collaborative Coping with IT Complexity by Combining SOA and Organic Computing. In: System and Information Sciences Notes, Vol.2, No.1. September 2007, S. 82—87.
- [VDE03] VDE/ITG/GI (Hrsg.): Positionspapier: Organic Computing - Computer- und Systemarchitektur im Jahr 2010. VDE/ITG/GI, 2003. Quelle: <http://surl.bsddgroup.de/61d66w> - letzter Abruf am 03.12.2007