Modularität in der verteilten Entwicklung komplexer Systeme: Chancen, Grenzen, Implikationen

Prof. Dr. Dres. h.c. Arnold Picot
Institutsleiter
Institut für Information, Organisation und Management
Munich School of Management
Ludwig-Maximilians-Universität München
Ludwigstr. 28
D-80539 München
Tel.: +49-(0)89-2180-2252
E-mail: picot@lmu.de

Dipl.-Wi.-Ing. Oliver Baumann, M.S. M.B.R.
Wissenschaftlicher Mitarbeiter und Doktorand
Institut für Information, Organisation und Management
Munich School of Management
Ludwig-Maximilians-Universität München
Ludwigstr. 28
D-80539 München
Tel.: +49-(0)89-2180-2982
E-mail: baumann@lmu.de

Wir danken den Herausgebern und zwei anonymen Gutachtern für wertvolle Kommentare zu einer früheren Version dieses Beitrags.

Modularität in der verteilten Entwicklung komplexer Systeme: Chancen, Grenzen, Implikationen

Zusammenfassung:

Die Entwicklung komplexer Systeme stellt Unternehmen vor die Herausforderung, das effektive Zusammenwirken einer Vielzahl interdependenter Elemente zu gewährleisten. Eine Systemdekomposition in Module mit klar definierten Schnittstellen gilt dabei als effiziente Designlösung, um dennoch die Vorteile von Spezialisierung und Arbeitsteilung zu nutzen. In jüngerer Zeit beschäftigen sich zahlreiche Arbeiten, insbesondere aus evolutionsökonomischer und komplexitätstheoretischer Perspektive, mit den prinzipiellen Grenzen und Implikationen der Modularität. Weil sich komplexe Systeme nicht perfekt zerlegen lassen und begrenzt rationale Designer nicht alle Interdependenzen überblicken, weisen modulare Systeme häufig unbekannte intermodulare Interdependenzen auf. Eine vollständig verteilte Entwicklung komplexer Systeme auf Basis unabhängiger Module ist deshalb nicht möglich und auch nicht sinnvoll. Die in vielen Bereichen steigende Komplexität macht ein fundiertes Verständnis der Vorteile, Grenzen und Implikationen modularer Ansätze in der verteilten Systementwicklung notwendig und bietet breites Potential für zukünftige Forschung.

Abstract:

In complex systems development, firms need to ensure the effective interplay of numerous interdependent elements. In order to still reap the benefits of specialization and division of labor, decomposing a system into modules with well-defined interfaces is considered an efficient design principle. However, a number of studies have recently started to explore the fundamental limits and implications of modularity from the perspectives of evolutionary economics and complexity theory. Because complex systems are at best “nearly decomposable” and because boundedly rational designers cannot account for all interdependencies, modular systems are never free from (potentially unknown) intermodular interdependencies that impede their fully autonomous and distributed development. As complexity is rising in various domains, a fundamental understanding of the contributions, limitations and implications of modularity in distributed systems development becomes necessary and offers broad opportunities for future research.

Schlagwörter: Modularität, Komplexität, Systemdesign und -entwicklung, Arbeitsteilung, Spezialisierung, Koordination

Keywords: Modularity, complexity, systems design and development, division of labor, specialization, coordination

JEL classification: L2, M1

1 Gestaltung komplexer Systeme als Managementherausforderung

Die Entwicklung leistungsfähiger komplexer Systeme stellt für Unternehmen in zahlreichen Industrien, etwa in der Softwarebranche, im Luft- und Raumfahrtsektor oder im Anlagenbau, eine wichtige Grundlage der Innovationsfähigkeit dar (Hobday et al. 2000). Die Frage, wie komplexe Systeme zu gestalten sind, um bestimmte Leistungsziele zu erreichen, ist jedoch eines der schwierigsten Probleme der Ingenieurwissenschaften wie auch der Betriebswirtschaftslehre (Simon 1996, Ulrich und Eppinger 2007). Weil komplexe Systeme aus einer Vielzahl interdependenter Elemente bestehen, müssen Systemhersteller insbesondere für das effektive Zusammenwirken (den „fit“) der Systembestandteile sorgen (Simon 1962, von Hippel 1990, Ulrich 1995). Dies ist allerdings keine rein technische Herausforderung, sondern betrifft gleichermaßen Aspekte der Organisation und des Managements. Die zur Koordination interdependenter Design- und Entwicklungsentscheidungen notwendige Abstimmung könnte erwarten lassen, dass Systemhersteller vor allem auf organisationsinterne Lösungen zurückgreifen, die eine unmittelbare Einflussnahme ermöglichen (Thompson 1967, Khandwalla 1977, Simon 1991). Doch das Gegenteil ist der Fall: Wie bereits in vielen produktionsnahen Tätigkeiten lässt sich auch bei wissensintensiven Prozessen ein branchenübergreifender Trend zu Spezialisierung und Arbeitsteilung beobachten und damit eine Zunahme der verteilten und häufig interorganisationalen Entwicklung komplexer Systeme (Hagedoorn 2002, Ernst 2005a, Eppinger und Chitkara 2006).

Die Gründe und Treiber dieser Entwicklung sind zahlreich: gesunkene Informations- und Kommunikationskosten (Picot et al. 1996, Picot et al. 2003), die Etablierung von Standards (Löwer 2006) oder die Notwendigkeit, externes Expertenwissen in Problemlösungsprozesse mit einzubeziehen (von Hippel 1994, Powell et al. 1996). Prinzipiell ermöglicht wird die arbeitsteilige Entwicklung komplexer Systeme jedoch erst durch die Nutzung modularer Produktarchitekturen. Modularität resultiert aus der Zerlegung (Dekomposition) eines Systems in einzelne Subsysteme (Module), deren gegenseitige Abhängigkeiten nach Möglichkeit eliminiert oder auf definierte Schnittstellen reduziert werden (Simon 1962, Alexander 1964, Baldwin und Clark 2000). Im Anschluss an eine solche Dekomposition lassen sich die einzelnen Module unabhängig voneinander entwickeln, etwa durch spezialisierte Zulieferer, und anschließend zu einem Gesamtsystem integrieren (Sanchez 1995, Langlois 2002). Aus diesem Grund haben sich modulare Ansätze in vielen Industrien – die Softwarebranche ist ein idealtypisches Beispiel – zu einem grundlegenden Designprinzip entwickelt (Baldwin und Clark 2000).

Die viel zitierten Vorteile modularer Strukturen haben jedoch prinzipielle Grenzen: Zum einen sind reale Systeme bestenfalls „nearly decomposable“ und lassen sich niemals in völlig unabhängige Module zerlegen (Simon 1996); zum anderen sind Systemdesigner nur begrenzt rational (Simon 1955, Simon 1956), weshalb Systemdekomposition und -entwicklung als typische Problemlösungsprozesse aufgefasst werden müssen, bei denen Systementwickler nach befriedigenden Designlösungen suchen müssen, also kein Optimum berechnen können (Simon 1955, March und Simon 1958, Cyert und March 1963). Modulare Dekompositionen komplexer Systeme weisen daher auch immer intermodulare Interdependenzen auf, die den Designern unbekannt sind oder von ihnen ignoriert werden (Ethiraj und Levinthal 2004, Staudenmeyer et al. 2005). Würde sich ein Systemhersteller streng am Kerngedanken der Modularität orientieren und die einzelnen Module völlig unabhängig voneinander entwickeln lassen, blieben diese intermodularen Abhängigkeiten unberücksichtigt. Optimierungsanstrengungen auf der Modulebene könnten sich dann negativ auf die Systemperformance auswirken und einen erhöhten Integrations- und Testaufwand nach sich ziehen. Aktuelle Entwicklungsprobleme bei großen komplexen Systemen wie TollCollect oder A380 illustrieren die Problematik.

Vor diesem Hintergrund hat sich in jüngerer Zeit eine Reihe von Arbeiten mit den Grenzen und Implikationen modularer Ansätze beschäftigt.[1] Diese Arbeiten sind im Wesentlichen zwei verschiedenen – wenngleich verwandten – Forschungssträngen zuzuordnen: Der erste Strang ist in der (empirischen) evolutionsökonomischen Literatur verankert. Die Autoren dieser Arbeiten weisen darauf hin, dass die verteilte Entwicklung in modularen Systemen nicht völlig unabhängig abläuft, sondern von sogenannten Systemintegratoren koordiniert werden muss (Brusoni und Prencipe 2001, Brusoni et al. 2001, Ernst und Kim 2002, Prencipe et al. 2003, Ernst 2005b, Hobday et al. 2005). Der zweite Forschungsstrang ist durch eine wachsende Zahl von Simulationsstudien gekennzeichnet, die mithilfe komplexitätstheoretischer Modelle den Zusammenhang von Organisationsstrukturen und verteilten Problemlösungsprozessen in interdependenten Entscheidungszusammenhängen untersuchen. Diese Studien haben sich etwa damit beschäftigt, wie Fehler bei der Systemdekomposition zu suboptimalen Pfadabhängigkeiten in der Systementwicklung führen können (Marengo et al. 2000, Ethiraj und Levinthal 2004, Marengo und Dosi 2005).

[1] Die Autoren haben in einem früheren, kleineren Beitrag einen Teil dieser Diskussion wiedergegeben (Picot und Baumann 2007).

Wir greifen die oben skizzierte Diskussion und die beiden Forschungsstränge auf und diskutieren im Folgenden auf abstrakter Ebene die Fragen:

1. Worin liegen die prinzipiellen Vorteile modularer Ansätze und welchen Beitrag leisten diese für die verteilte Entwicklung komplexer Systeme?

2. Worin liegen die prinzipiellen Grenzen der Modularität und welche Implikationen ergeben sich daraus für die verteilte Systementwicklung?

Aus der Beantwortung dieser Fragen lassen sich dann auch erste Folgerungen für die Koordination verteilter Entwicklungsvorhaben, etwa durch einen Projektkoordinator oder Systemhersteller, ziehen.

Der Beitrag vertritt eine positive, aber nicht uneingeschränkt kritikfreie Haltung gegenüber dem Nutzen modularer Ansätze. Modularität ist ein mächtiges Strukturprinzip und stellt eine notwendige Voraussetzung dar, um auch bei der Entwicklung komplexer Systeme die Vorteile von Spezialisierung und Arbeitsteilung nutzbar zu machen. Wir weisen jedoch auch darauf hin, dass modulares Design nicht zu einer „vanishing hand of management“ (Dosi et al. 2003, S. 109) in Form eines reinen „plug and plays“ unabhängig voneinander entwickelter Module führen kann, wie es zum Teil propagiert wird (Sanchez 1995, Langlois 2002, Langlois 2003). Dies wird auch mit der Vision einer „Lego economy“ illustriert, die die Effizienz und Flexibilität ihrer komplexen Systemstrukturen aus vielen simpel zusammensteckbaren Komponenten bezieht (Hagel und Brown 2001, Veryard 2001). In komplexen Entwicklungsvorhaben bleibt vielmehr die „visible hand“ (Dosi et al. 2003, S. 109) des Systemherstellers unerlässlich für die Systemdekomposition, die Koordination der Modulentwickler und die Integration der Module zu einem leistungsfähigen Gesamtsystem. Die wachsende Komplexität vieler Systeme und deren zunehmend arbeitsteilige Entwicklung basierend auf modularen Strukturen lassen erwarten, dass die Bedeutung derartiger Fähigkeiten in Zukunft gleichermaßen zunehmen wird. Um angemessen mit dieser Entwicklung umgehen zu können, ist ein systematisches und fundiertes Verständnis der Vorteile, Grenzen und Implikationen modularer Ansätze notwendig.

Der Beitrag ist folgendermaßen gegliedert: Abschnitt zwei geht auf die charakteristischen Eigenschaften komplexer Systeme ein, die im Rahmen von Design- und Entwicklungsbemühungen von Bedeutung sind. Abschnitt drei greift die erste Forschungsfrage auf und fasst die Kerngedanken modularen Designs und dessen Beitrag für die verteilte Entwicklung komple-

xer Systeme zusammen. Die Abschnitte vier und fünf befassen sich mit der zweiten Forschungsfrage: Zunächst werden die prinzipiellen Grenzen der Modularität in komplexen Systemen behandelt, im Anschluss wesentliche empirische Arbeiten und Simulationsstudien zusammengefasst, die sich mit den Implikationen dieser Grenzen für verteilte Entwicklungsvorhaben befasst haben. Der letzte Abschnitt zieht ein Fazit und skizziert viel versprechende Felder für die zukünftige Forschung im Spannungsfeld von Modularität und der verteilten Entwicklung komplexer Systeme.

2 Charakteristische Eigenschaften komplexer Systeme

Unser Verständnis von einem "komplexen System" in diesem Beitrag basiert auf Simons abstrakter Definition: „[B]y a complex system I mean one made up of a large number of parts that have many interactions. … in such systems the whole is more than the sum of the parts in the weak but important pragmatic sense that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole" (Simon 1996, S. 183f). Komplexe Systeme bestehen demnach aus einer Vielzahl an Elementen, die auf nichttriviale Weise interagieren. Ihre Komplexität resultiert dabei vor allem aus der Anzahl, Art und Stärke der Interdependenzen zwischen den einzelnen Elementen und aus den Auswirkungen dieser Beziehungen auf das Gesamtsystem. Mit „Interdependenz" bezeichnen wir dabei eine generelle Form von Interaktion oder Abhängigkeitsbeziehung zwischen zwei Elementen, die sowohl positiv (zwei Elemente verstärken sich in ihrer Wirkung), negativ (die Elemente schwächen sich in ihrer Wirkung ab) oder auch in beide Richtungen wirksam sein kann. Gleichzeitig können sich Stärke und Richtung von Interdependenzen für konkrete Implementierungen der beteiligten Systemelemente ändern (Eppinger et al. 1994, Ulrich und Eppinger 2007). Komplexe Systeme lassen sich somit auch umso schwerer „überblicken" und in ihrem Verhalten beschreiben, je mehr Elemente und Interdependenzen sie aufweisen (Bar-Yam 1997). In dem Maße wie Dynamik auftritt, also sich z.B. Elemente in einem System oder externe Einflüsse auf das System verändern, verschärft sich die Problematik der Komplexität.

Je komplexer ein System ist, desto stärker vernetzt sind daher seine Elemente. Mit steigender Vernetzung nimmt jedoch auch die Korrelation eines Systems ab, also das Maß dafür, inwiefern ähnliche Konfigurationen der Systemelemente auch zu einer ähnlichen Systemleistung führen (Kauffman 1993, Kauffman 1995). Dies hat zur Folge, dass Änderungen an einzelnen Elementen des Systems häufig nicht lineare, sondern vielmehr nichtlineare Verhaltens- bzw. Leistungsänderungen des Gesamtsystems nach sich ziehen. Bei der Entwicklung komplexer

Systeme führen lokale Optimierungsanstrengungen, d.h. die Suche nach besseren Lösungen auf der Subsystemebene, deshalb oft zu unerwünschten, nichtlinearen Effekten auf der Systemebene und damit nicht automatisch zu einer globalen Verbesserung. In anderen Worten: Innovation in einem Teil des Systems verursacht häufig Probleme in anderen Teilen. Das Problem ähnelt dem der Gestaltung der Unternehmensorganisation. Auch dabei bedient man sich in der Regel eigenständiger Subsysteme (funktionale Abteilungen, Geschäfts-, Zentralbereiche usw.) mit der Gefahr einer suboptimalen Gesamtlösung trotz (oder gerade wegen der) Perfektionierung von Teilbereichen.

Weiterhin steigt mit einem höheren Maß an Interdependenz zwischen den Elementen eines Systems auch die Zahl der lokalen Optima, also jener möglicher Anordnungen der Systemelemente, die nicht mehr ohne Weiteres verbessert werden können, da bereits geringfügige Änderungen einzelner Teile zu unvorhergesehenen Effekten und starken Leistungseinbußen führen (Kauffman 1993, Kauffman 1995). Zur weiteren Verbesserung der Systemleistung wäre in solchen Fällen die schwer zu überblickende und damit risikoreiche *gleichzeitige* Veränderung vieler Systemelemente notwendig. Die Wahrscheinlichkeit ist daher groß, dass in der Systementwicklung ein einmal erreichtes lokales Optimum nicht mehr aufgeben wird, auf die Gefahr hin, dass noch weitaus bessere Alternativen (überlegene lokale Lösungen) existieren. Diese Eigenschaften komplexer Systeme machen effektive Design- und Entwicklungsanstrengungen daher kompliziert und fehleranfällig.

3 Bedeutung modularer Ansätze für die Systementwicklung

3.1 Komplexitätsmanagement durch Systemdekomposition

Die Verwendung einer modularen Systemarchitektur anstelle eines integrierten Designs, also anstelle der Entwicklung eines komplexen Systems in seiner Gesamtheit, kann helfen, die Komplexität beherrschbar zu machen: Durch Bündelung einzelner Systemelemente zu Modulen und deren anschließender Entwicklung lässt sich die Zahl der gleichzeitig zu betrachtenden Variablen auf ein überschaubares Maß reduzieren. Dafür muss zunächst eine Dekomposition des zu entwickelnden Systems erarbeitet werden, d.h. eine Zerlegung in Module sowie eine Zuordnung der Systemelemente zu den einzelnen Modulen. Eine wesentliche Eigenschaft modularer Dekompositionen ist dabei die starke Kopplung innerhalb einzelner Module (also ein hohes Maß an intramodularer Interdependenz) im Gegensatz zu der nur schwachen Kopplung zwischen den Modulen (ein geringes Maß an intermodulare Interdependenz) (Simon 1962, Alexander 1964, Weick 1976, Orton und Weick 1990, Baldwin und Clark

2000). Um stark interdependente Elemente in einem gemeinsamen Modul zusammenzufassen und gleichzeitig die Interdependenzen zwischen den einzelnen Modulen zu minimieren, lassen sich (iterativ) eine Reihe von Dekompositionsansätzen und -verfahren einsetzen (Parnas 1972, Baldwin und Clark 2000). Abgesehen von der Entwicklung komplexer technischer Systeme lassen sich Tendenzen zu einer derartigen Dekomposition großer Systeme in einzelne Module auch in anderen unternehmensrelevanten Zusammenhängen beobachten, wenngleich teilweise unter anderem Namen. Dies gilt in prinzipieller Weise etwa für die M-Form, also die Zusammenfassung stark interdependenter Entscheidungszusammenhänge zu Unternehmensfunktionen (Thompson 1967, Galbraith 1973, Williamson 1975, Burton und Obel 1998), aber auch für die weitergehende Organisationsgestaltung nach dem Prinzip lose gekoppelter Einheiten (Weick 1976, Picot und Freudenberg 1998, Schilling und Steensma 2001, Langlois 2002, Picot et al. 2003).

Liegt eine modulare Dekomposition eines komplexen Systems vor, wird nach „außen“ hin nur die abstrakte Funktion bekannt gemacht, die ein Modul im Kontext des Gesamtsystems übernimmt, sowie seine Schnittstelle, also die Art und Weise, wie sich das Modul ansprechen lässt (Baldwin und Clark 2000, Schilling 2000, Langlois 2002). Die Kommunikation mit den anderen Modulen findet dann über solche standardisierten Schnittstellen statt. Ein Großteil der eigentlichen Komplexität, die sich aus der Zahl und dem Interaktionsverhalten der Modulelemente ergibt, wird dagegen innerhalb der Module „versteckt“ und ist nur für die jeweiligen Entwickler des Moduls sichtbar und von Interesse (Baldwin und Clark 2000). Derartige modulare Strukturen und die Kopplung der Module über genormte Schnittstellen ermöglichen es daher, die Komplexität eines Systems auf ein beherrschbares Maß „herunter zu brechen“. Das Vorgehen gleicht im Grundsatz dem Konzept der objektorientierten Programmierung bzw. der komponenten- oder serviceorientierten Architekturen bei der Entwicklung von Informationssystemen. Auch dabei werden relativ autonome Objekte, Komponenten oder Services als Elemente eines größeren, in der Regel komplexen Systems abgegrenzt und über standardisierte Schnittstellen verknüpft, so dass sich ihre Binnenfunktionalität nur über definierte Interaktionen wie Nachrichten oder Aufträge aktivieren lässt (Booch et al. 2007, Erl 2008).

3.2 Flexibilisierung und Beschleunigung von Entwicklungsprozessen

Zunächst stand der skizzierte Beitrag zur Komplexitätsreduktion im Vordergrund der wissenschaftlichen Diskussion um die Modularisierung. In jüngerer Zeit beschäftigt sich die Literatur vor allem mit den positiven Auswirkungen modularer Systeme auf Innovationsprozesse.

Unter dem Begriff der „power of modularity“ (Baldwin und Clark 2000) werden verschiedene Konzepte postuliert, die sich auf die Vereinfachung, Optimierung und Flexibilisierung von Design- und Entwicklungstätigkeiten beziehen. Die Gemeinsamkeit dieser Konzepte besteht darin, dass Modularität auch bei kognitiven Tätigkeiten, wie sie die Systementwicklung in besonderem Maße darstellt, ein arbeitsteiliges Vorgehen ermöglicht. Modulare Systeme sind demnach die Voraussetzung zur „autonomen Innovation“, d.h. zur unabhängigen Entwicklung individueller Module (Henderson und Clark 1990): Weil für das übrige System nur die jeweilige Funktion ausschlaggebend ist, die ein Modul erfüllt, und diese über eine standardisierte Schnittstelle zur Verfügung gestellt wird, kann das (zumeist komplexe) Innenleben des Moduls autonom entwickelt, verändert oder verbessert werden, ohne dass dies Auswirkungen auf andere Systembereiche hätte. Modulübergreifende Koordinationsbemühungen werden in diesem Prozess nicht benötigt (Sanchez 1995, Langlois 2002). Aufgrund ihrer Unabhängigkeit können die einzelnen Module außerdem parallel entwickelt werden. Dadurch lassen sich kürzere Entwurfs-, Entwicklungs- und Testzeiten realisieren als bei einem integrierten Vorgehen, was insbesondere in Industrien mit schnellen Innovationszyklen einen entscheidenden strategischen Vorteil darstellen kann (Baldwin und Clark 2000, Loch et al. 2001, Marengo und Dosi 2005, Ulrich und Eppinger 2007).

Ein weiterer Vorteil wird durch den Begriff der „modularen Innovation“ überschrieben (Henderson und Clark 1990, Göpfert 1998, Baldwin und Clark 2000). Dieser bezieht sich auf die Möglichkeit, Module aufgrund ihrer jeweiligen Funktion im Baukastenverfahren zu kombinieren („mix and match“) und damit auf einfache Weise neue Systemfunktionalitäten oder Designvarianten zu realisieren. Das Konzept der modularen Innovation bezieht sich außerdem auf die Möglichkeit, mit einzelnen Modulen zu experimentieren: Wenn die Module in unterschiedlichen Designvarianten entwickelt werden, so lässt sich ein „best of breed approach“ verfolgen, indem das jeweils leistungsfähigste Modul ausgewählt und in das System integriert wird (Baldwin und Clark 2000). Je mehr Module ein System aufweist und je mehr Designvarianten für die einzelnen Module geschaffen werden, also je mehr mit den Modulen experimentiert wird, desto einfacher und schneller können weniger gute Modulvarianten durch leistungsfähigere ersetzt werden. Diese Möglichkeit, Veränderungen und Verbesserungen einzelner Module vornehmen zu können (aber nicht zu müssen), erhöht den Optionswert einer Produktarchitektur signifikant (Baldwin und Clark 2000). Mittels modularer Architekturen lassen sich komplexe Systeme somit unabhängiger und flexibler gestalten sowie für zukünftige Änderungen und Erweiterungen vorbereiten (Ulrich 1995).

3.3 Beitrag zur verteilten Systementwicklung

Eine modulare Systemarchitektur stellt daher einen entscheidenden Faktor dar, um die Vorteile von Spezialisierung und Arbeitsteilung auch bei Problemlösungsprozessen wie der Entwicklung komplexer Systeme zu nutzen: Sobald die Dekomposition eines geplanten Systems vorliegt und die Funktionen und Schnittstellen der einzelnen Module definiert sind, kann – basierend auf dem Konzept der „autonomen Innovation" – die weitere Systementwicklung im Idealfall unabhängig und über verschiedene Standorte verteilt ablaufen. Gleichzeitig lassen sich Systeme – basierend auf dem Konzept der „modularen Innovation" – aber auch aus verteilt entwickelten und bereits existierenden Modulen zusammensetzen. Eine solche Variantenbildung in Reinform lässt sich typischerweise in der PC-Industrie beobachten (Baldwin und Clark 2000). Aber auch in der Softwareindustrie gibt es angesichts einer zunehmend globalen und industriemäßigen Systementwicklung derzeit ähnliche Tendenzen hin zu einer stärkeren Standardisierung von Funktionalitäten in Modulen oder Modul-ähnlichen Konstrukten, die sich in immer neuen Kombinationen einsetzen lassen (Löwer 2006, Erl 2008).

Zu einer unternehmensübergreifenden Entwicklung komplexer Systeme kommt es im allgemeinen etwa dann, wenn sich Systemhersteller auf zentrale Subsysteme konzentrieren und die Entwicklung von nicht-kritischen und stärker standardisierten (oder standardisierbaren) Modulen an spezialisierte Zulieferer abgeben (Schilling und Steensma 2001, Hoetker et al. 2007). Bei der Entwicklung komplexer Systeme, in denen meist eine breite Palette unterschiedlicher Technologien zum Einsatz kommt, ist es häufig sogar unumgänglich, einzelne Subsysteme unternehmensextern zu beziehen, weil dem Systemhersteller das notwendige spezifische Wissen oder andere Ressourcen nicht unmittelbar zur Verfügung stehen (Pisano 1990, von Hippel 1994, Powell et al. 1996). Eine verteilte Entwicklung komplexer modularisierter Systeme lässt sich aber auch unter anderen Ausgangsbedingungen beobachten: Angesichts des oftmals lukrativeren und nachhaltigeren Geschäfts mit der Bereitstellung von Services und Solutions im Umfeld eines Systems treten Systemhersteller häufig nur noch als Systemarchitekten und -integratoren auf. Die eigentliche Entwicklungsarbeit auf Modulebene wird dann zu einem großen Teil an spezialisierte Zulieferer abgegeben (Hobday et al. 2000, Pavitt 2003, Hobday et al. 2005).

Im Kontext von Modularität und verteilter Systementwicklung lässt sich außerdem eine gewisse Affinität zwischen der Architektur eines Systems und der Organisationsform beobachten, die bei seiner Entwicklung zum Einsatz kommt (Henderson und Clark 1990, Göpfert

1998, Sako 2003). Dieser Zusammenhang wird dadurch erklärt, dass sich Produktarchitekturen allmählich ihren Entwicklungsumgebungen anpassen oder diese aufgrund von konkreten Managemententscheidungen reflektieren, und dass sich, im Falle statischer Produktarchitekturen, langfristig auch Organisationsstrukturen entsprechend verändern (Takeishi 2001, Langlois 2002, MacCormack et al. 2006). Eine aktuelle Entwicklung in der Softwarebranche verdeutlicht diesen Zusammenhang exemplarisch: Im Rahmen von Open Source Softwareprojekten, also bei Softwaresystemen, deren Quellcode frei zugänglich ist und an deren Entwicklung jeder interessierte Programmierer weltweit mitwirken kann, findet die Systementwicklung häufig (stark) verteilt statt (Brügge et al. 2004). Und obgleich die These wohl nicht auf alle Open Source Projekte verallgemeinerbar ist, hat beispielsweise eine Analyse des Quellcodes gezeigt, dass das Open Source Betriebssystem Linux, das verteilt entwickelt wurde, einen höheren Modularitätsgrad aufweist als der Webbrowser Mozilla, der proprietär und nicht verteilt entwickelt wurde (MacCormack et al. 2006). In diesem Zusammenhang lautet daher die These, dass eine stärkere Modularisierung notwendig sei, damit die verteilte Systementwicklung überhaupt stattfinden könne (O'Reilly 1999). Nur durch eine größere Zahl an eher kleineren Modulen werde eine „architecture for participation“ (Baldwin und Clark 2006, MacCormack et al. 2006) geschaffen, die es einer Vielzahl an freiwilligen Entwicklern ermöglicht, einen für eine Einzelperson überschaubaren und sinnvollen Teil zur Entwicklung eines Gesamtsystems beizutragen.

4 Grenzen der Modularität in komplexen Systemen

Die oben skizzierten Konzepte unterstreichen die Bedeutung der Modularität für die Effizienz und Flexibilität der (verteilten) Entwicklung komplexer Systeme. Trotz (oder gerade wegen) der großen Popularität modularer Systeme in der akademischen und unternehmenspraktischen Diskussion beschäftigt sich seit einiger Zeit jedoch auch eine wachsende Zahl an Arbeiten mit deren prinzipiellen Grenzen und Implikationen für Fragen der Organisation, Innovation und Unternehmensstrategie (siehe etwa Marengo et al. 2000, Brusoni und Prencipe 2001, Brusoni et al. 2001, Dosi et al. 2003, Fleming und Sorenson 2003, Pavitt 2003, Sako 2003, Ethiraj und Levinthal 2004, Ernst 2005b, Marengo und Dosi 2005). Ohne die Vorteile der Modularität prinzipiell in Frage zu stellen, äußern diese Arbeiten dabei die Kritik, dass die bisherige wissenschaftliche Auseinandersetzung (bewusst oder unbewusst) wesentliche Aspekte der Realität vernachlässigt habe.

Stattdessen rücken sie zwei grundlegende Faktoren in den Vordergrund, die dem idealisierten Bild von der simplen Dekomposition komplexer Systeme und der anschließenden unabhängigen Modulentwicklung, das vielen Arbeiten zur Modularität (teilweise implizit) zugrunde liegt, fundamentale Grenzen setzen: Zum einen ist dies die Tatsache, dass reale Systeme bestenfalls „nearly decomposable“ sind und sich niemals in völlig unabhängige Module zerlegen lassen (Simon 1996). Ein gewisses Maß an intermodularer Interdependenz bleibt somit unvermeidbar und muss bei der verteilten Modulentwicklung beachtet werden, um unerwünschte Effekte auf der Systemebene zu vermeiden. Zum anderen ist es das Konzept der beschränkten Rationalität (Simon 1955, Simon 1956) und das verhaltenswissenschaftliche Verständnis von Problemlösungsprozessen als adaptiven Such- und Entscheidungsprozessen (March und Simon 1958, Cyert und March 1963, Newell und Simon 1972, Nelson und Winter 1982). Anstelle einer Optimierung durch allwissende Designer lassen sich Systemdekomposition und Modulentwicklung daher vielmehr als evolutionäre (und damit als pfadabhängige, unvollkommene und fehleranfällige) Prozesse modellieren, bei denen die Systementwickler mit einem großen (und für sie meist unüberschaubaren) Bündel interdependenter Designvariablen konfrontiert sind, für das sie nach einer guten Lösung suchen müssen: „‘[G]ood’ designs, modular or otherwise, are a product of evolution rather than foresight“ (Ethiraj und Levinthal 2004, S. 160). Im Folgenden gehen wir näher auf diese fundamentalen Beschränkungen ein.

4.1 Die Eigenschaft der “near decomposability“

Eine idealtypische Systemdekomposition würde starke intramodulare und extrem wenige (bzw. überhaupt keine) intermodulare Interdependenzen aufweisen. Solche Dekompositionen stellen wohl einen theoretischen Grenzfall dar und sind in realen Systemzusammenhängen praktisch nicht beobachtbar. Viele Systeme können nach Simon jedoch als „nearly decomposable“ bezeichnet werden (Simon 1996, Simon 2002). Die Eigenschaft der „near decomposability“ bedeutet, dass eine Systemzerlegung möglich ist, bei der die Zahl der Interdependenzen innerhalb der einzelnen Module relativ hoch ist, während die Module untereinander nur schwach gekoppelt sind. Ein reales System muss dabei nicht von Anfang an eine solche Struktur aufweisen. Vielmehr lassen sich die Interdependenzen in komplexen, integrierten Systemen mithilfe verschiedener Designverfahren (etwa über die Definition von „design rules“, also von Schnittstellen und Standards) meist derartig beeinflussen, dass sich daraus ein zerlegbares System im Sinne der „near decomposability“ erzeugen lässt (Parnas 1972, Baldwin und Clark 2000).

Zur Illustration und weiteren Analyse dieser Zusammenhänge beschreiben wir die Interdependenzen in einem System auf abstrakte Weise. Angenommen, bei der Systementwicklung seien insgesamt *n* Designvariablen zu berücksichtigen, für die jeweils eine Entscheidung bezüglich ihrer Umsetzung getroffen werden muss. In der Automobilentwicklung ließe sich als Designvariable etwa die Wahl einer bestimmten Motorleistung oder die Auslegung des Bremssystems vorstellen. Abb. 1 zeigt ein Beispiel, wie sich dann mit Hilfe so genannter Interaktionsmatrizen die Abhängigkeitsbeziehungen zwischen den Designvariablen eines Systems und dessen modularer Zerlegung grafisch darstellen lassen.[2] Das abgebildete System besteht aus *18* Designvariablen, die zu drei Modulen ($a_1, \ldots, a_6$; $b_1, \ldots, b_6$; $c_1, \ldots, c_6$) zusammengefasst wurden. Ein „*x*" in einer Zeile bedeutet, dass die jeweilige Zeilenvariable durch die entsprechende Spaltenvariable beeinflusst wird. Im obigen Automobilbeispiel könnte ein „x" also etwa bedeuten, dass sich die Wahl einer bestimmten Motorleistung auf die optimale Auslegung des Bremssystems auswirkt.

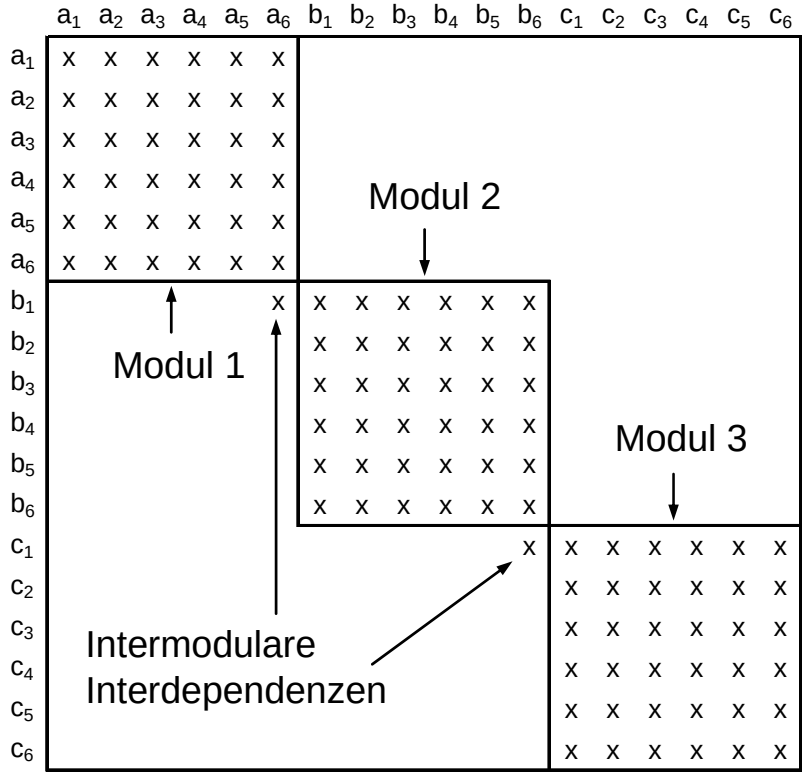

Abb. 1: Interaktionsmatrix eines „nearly decomposable systems" mit N=18 Designvariablen und M=3 Modulen (in Anlehnung an: Ethiraj und Levinthal 2004)

[2] In der Literatur zum Design komplexer Systeme findet sich eine ähnliche Darstellung unter dem Konzept der „Design Structure Matrix" (Steward 1981, Eppinger et al. 1994), das zur Darstellung, Analyse und Gruppierung von Abhängigkeiten zwischen Bauteilen oder Subsystemen eingesetzt wird.

In Abb. 1 liegen größtenteils reziproke Interdependenzen zwischen jeweils zwei Designvariablen vor, was sich an einer symmetrischen Kennzeichnung mit einem „*x*“ oberhalb und unterhalb der Diagonalen erkennen lässt. Die angedeutete Dekomposition stellt außerdem eine gute modulare Zerlegung dar, weil die Designvariablen innerhalb der drei Module jeweils eng miteinander gekoppelt sind. Die einzelnen Module sind allerdings nicht völlig unabhängig voneinander, denn es besteht je eine Kopplung zwischen dem ersten und dem zweiten Modul (b_1 wird durch a_6 beeinflusst) sowie zwischen dem zweiten und dem dritten Modul (c_1 wird durch b_6 beeinflusst). Aufgrund dieser intermodularen Interdependenzen ist das System deshalb nur „nearly decomposable“. Das System ist darüber hinaus hierarchisch (Simon 1996): Das zweite Modul hängt (über b_1) von den Entscheidungen ab, die bezüglich der Variablen im ersten Modul getroffen wurden; umgekehrt besteht jedoch keine Abhängigkeit. Gleichermaßen hängt das dritte Modul (über c_1 sowie indirekt über b_1) von den Entscheidungen ab, die bezüglich der Variablen im zweiten und im ersten Modul getroffen wurden, während in der „Gegenrichtung“ keine Abhängigkeiten bestehen. Die dargestellte Interaktionsstruktur repräsentiert somit eine Abhängigkeitshierarchie, mit dem ersten Modul an der Spitze und dem dritten Modul am unteren Ende. Die dargestellte modulare Systemzerlegung ist dennoch „optimal“ in der Hinsicht, dass sie der zugrunde liegenden tatsächlichen Interaktionsstruktur bestmöglich entspricht. Die prinzipielle Schwierigkeit liegt allerdings darin, dass eine solche optimale Zerlegung komplexer Systeme in der Regel nicht offensichtlich ist, sondern zunächst entwickelt werden muss. Diese Aufgabe ist auch keineswegs trivial, sondern stellt ein NP-vollständiges Problem dar (Schaefer 1999, Chapman et al. 2001), das für große Systeme damit praktisch nicht optimal lösbar ist.[3]

4.2 Systemdekomposition und Systementwicklung als evolutionäre Problemlösungsprozesse

Selbst im Fall eines perfekt zerlegbaren Systems wäre es unwahrscheinlich, dass es begrenzt rationalen Systemdesignern gelingen könnte, alle Elemente und Abhängigkeitsbeziehungen eines großen Systems vollständig und adäquat zu beschreiben. Designer verfügen bestenfalls über (vereinfachende) mentale Modelle der Systemzusammenhänge, so genannte kognitive Repräsentationen (March und Simon 1958, Holland et al. 1986, Tversky und Kahneman 1986,

[3] Der Begriff der NP-Vollständigkeit stammt aus der Komplexitätstheorie. Er bezeichnet eine Klasse von Problemen, für die keine deterministischen Algorithmen bekannt sind, mit denen sie in polynomineller Zeit lösbar wären. Stattdessen steigt bei den bekannten Algorithmen die Zahl der für die Problemlösung notwendigen Schritte (und damit die Rechenzeit) exponentiell mit der Problemgröße. Zur näherungsweisen Lösung derartiger Probleme werden daher vereinfachende Verfahren (Heuristiken) eingesetzt.

Simon 1991, Gavetti und Levinthal 2000) oder über erfahrungsbasierte Kenntnisse in lokalen Systembereichen (Cohen und Levinthal 1989, Cohen und Levinthal 1990, March 1991, Levinthal und March 1993). Sie können deshalb keine optimalen Systemdekompositionen „berechnen", sondern müssen vielmehr nach einer „angemessen guten" Dekompositionslösung suchen. Nach Simon entspricht dies einer so genannten „satisificing"-Verhaltensweise (zusammengesetzt aus „to satisfy" und „sufficient"; steht im Gegensatz zum neoklassischen Optimierungskalkül) (Simon 1955, Simon 1956). Diese Suchbemühungen erstrecken sich etwa (1) auf eine „angemessene" Anzahl an Modulen, (2) auf eine „angemessene" Aufteilung der Designvariablen auf die Module, (3) auf „angemessene" Interdependenzen innerhalb der Designelemente der einzelnen Module sowie (4) auf „angemessene" Interdependenzen zwischen den Modulen (Ethiraj und Levinthal 2004).

Dabei können einerseits unterstützende, (semi-)formale Dekompositionsheuristiken zum Einsatz kommen, etwa zur Beschreibung von Produktfunktionen und Schnittstellen oder zur Identifikation von Funktions- und Komponentenclustern (Baldwin und Clark 2000, Wenzel 2003, Ulrich und Eppinger 2007); andererseits bleibt der diskretionäre Entscheidungsspielraum der Systemdesigner weiterhin groß. Eine als „angemessen" erachtete Dekomposition stellt daher letztlich nur eine (mehr oder weniger gute) „Vermutung" der Designer über die tatsächliche Interaktionsstruktur eines Systems dar. Die Wahrscheinlichkeit ist daher hoch, dass bei der Dekomposition eines komplexen Systems eine zu hohe („overmodular decomposition") oder zu niedrige („undermodular decomposition") Zahl an Modulen gewählt wird, als sie eigentlich angebracht wäre, oder dass die Designelemente suboptimal auf die einzelnen Module verteilt werden (Ethiraj und Levinthal 2004).[4] In Konsequenz führt dies dazu, dass trotz aller Dekompositionsbemühungen immer Interdependenzen zwischen den Modulen eines Systems bestehen bleiben, die auch den Systemdesignern oft nur zum Teil bekannt sind.

Auch die eigentliche Systementwicklung ist als typischer Problemlösungsprozess von einem hohen Maß an Unsicherheit geprägt und damit bestenfalls in Ansätzen berechen- und standardisierbar. Systementwicklung durch begrenzt rationale Akteure lässt sich daher als ein evolutionärer Such- und Entscheidungsprozess verstehen, der darauf abzielt, neue Designvarianten

[4] Darüber hinaus wird die konkrete Modularisierung eines Systems in der Realität auch durch die unterschiedlichsten Randbedingungen bestimmt, denen ein Systemhersteller unterworfen sein kann. Dies ist zum Beispiel bei sogenannten Legacy-Systeme der Fall – also bei bestehenden Altsystemen oder -systembestandteilen, die in den Systementwurf mit einbezogen werden müssen – oder auch bei bedeutenden Modullieferanten, an denen sich der übrige Systementwurf orientieren muss.

zu identifizieren, die eine höhere Systemperformance ermöglichen als die jeweils bisherige Lösung (Schumpeter 1939, Cyert und March 1963, Levinthal und March 1982, Nelson und Winter 1982). Die evolutionäre Logik äußert sich dabei vereinfacht folgendermaßen: Durch Änderung der aktuellen Systemkonfiguration wird eine neue Designvariante generiert (Variation) und die daraus resultierende Performance im Vergleich zur bisherigen Lösung evaluiert (Selektion). Im Anschluss wird die leistungsfähigere Alternative beibehalten und zur Grundlage der weiteren Suchbemühungen gemacht (Retention).

Die begrenzte Rationalität der Systemdesigner, die nach guten Lösungen suchen müssen anstatt sie zu „berechnen", haben dabei einen großen Einfluss darauf, welche Designalternativen überhaupt erzeugt und selektiert werden können. Suchprozesse basieren dabei sowohl auf „online search", also auf der tatsächlichen Umsetzung einer Idee und dem daraus resultierenden erfahrungsbasierten Lernen (trial and error learning) (Cyert und March 1963, Nelson und Winter 1982), aber auch auf „offline search", d.h. auf unterstützenden kognitiven Prozessen, etwa im Rahmen von Simulationen, Berechnungen oder ähnlichen Methoden (March und Simon 1958, Holland et al. 1986, Tversky und Kahneman 1986, Simon 1991, Gavetti und Levinthal 2000). Systemdesigner gehen dabei jedoch meist von ihren bestehenden Kompetenzen und Wissensbeständen aus und suchen vorwiegend „lokal" nach Verbesserungen, d.h. in einem begrenzten technologischen „Umkreis" um ihre aktuelle Konfiguration, nicht jedoch auf radikal anderen Gebieten (Cyert und March 1963, Levinthal und March 1982, Nelson und Winter 1982, Cohen und Levinthal 1989, Levinthal und March 1993, Stuart und Podolny 1996, Fleming und Sorenson 2003). Die Gründe für derartige lokale Suchprozesse liegen zum einen in dem großen Aufwand, der mit dem Aufbau des entsprechenden Wissens und der jeweiligen Kompetenzen für weiter entfernte Gebiete verbunden ist, zum anderen auch in den hohen Risiken, die eine Suche in unsicheren, „fremden" Gebieten mit sich brächte. Systemdesigner „ertasten" mittels kleiner Schritte in lokaler Suche deshalb so lange Leistungsverbesserungen, bis sie eine Systemkonfiguration entwickelt haben, die sich durch zusätzliche inkrementelle Änderungen nicht weiter verbessern lässt (Simon 1996). Die Modul- und Systementwicklung ist deshalb in vielen Fällen ein pfadabhängiger Lernprozess, bei dem frühere Varianten großen Einfluss darauf haben, welche neuen Konfigurationen aktuell und in Zukunft gefunden werden. Die Effizienz solcher lokaler Suchprozesse hängt dabei von der Komplexität des Systems ab: Wie in Abschnitt zwei dargestellt, erhöht sich mit der Zahl der Interdependenzen zwischen den Designvariablen eines Systems auch die Zahl der lokalen Optima, also jener Konfigurationen, die sich durch lokale Veränderungen nicht weiter verbessern las-

sen. Designer können bei der Entwicklung komplexer Systeme daher schnell auf Systemkonfigurationen stoßen, die sich durch lokale Suchbemühungen nicht weiter verbesern lassen (Kauffman 1993, Kauffman 1995, Levinthal 1997).

Die Implikationen dieser Art des Vorgehens für die verteilte Entwicklung modularer Systeme lassen sich wieder anhand des oben eingeführten abstrakten Beispiels verdeutlichen. Es sei angenommen (ohne Beschränkung der Allgemeinheit), für jede der *n* Designvariablen existierten jeweils zwei Designalternativen. Damit ließen sich theoretisch 2^n mögliche Designvarianten für das Gesamtsystem realisieren. Diesem „design space", also der Menge der möglichen Konfigurationen, stehen Systemdesigner in ihrer Suche nach Verbesserungen gegenüber (Simon 1996). Mit einer zunehmenden Zahl an Designvariablen und deren Ausprägungen wächst dieser Raum jedoch schnell so stark, dass eine vollständige Suche über alle Konfigurationen unmöglich wird. Modulare Systeme weisen bezüglich dieser Suchprozesse einige Besonderheiten auf: Zum einen lässt sich die Suche jeweils auf eine Modulebene begrenzen, was die individuelle Verbesserung einzelner Module ermöglicht (autonome Innovation). Aufgrund der Dekomposition des Gesamtsystems wird der Suchraum für die Modulentwickler reduziert, was eine schnellere und/oder bessere Optimierung der einzelnen Module ermöglichen kann. Weiterhin erlauben modulare Systeme die Rekombination von Modulen, also das Austauschen einzelner Module durch funktional äquivalente, aber leistungsfähigere Module (modulare Innovation).

Weil jedoch in modularen Systemen noch immer (teilweise unbekannte) Interdependenzen zwischen den einzelnen Modulen bestehen, lässt sich durch die verteilte und unabhängige Optimierung der einzelnen Module, sei es durch lokale Suche auf Modulebene oder durch Rekombination, in der Regel keine globale Optimierung des Systems erreichen. Angenommen, jede der *n* Designvariablen leiste einen gewissen unterscheidbaren Beitrag zur Gesamtperformance des Systems. Ist in diesem Modell eine Designvariable von allen anderen Variablen unabhängig, so hängt ihr Performancebeitrag nur davon ab, wie sich die Systemdesigner bezüglich dieser Variablen entscheiden. Eine aus Sicht einer speziellen Variablen „optimale" Entscheidung ist damit auch optimal mit Blick auf das Gesamtsystem. Bestehen jedoch Interaktionen zwischen den Designvariablen, so hängt der Beitrag zur Gesamtperformance, den eine Designentscheidung für Variable *x* mit sich bringt, auch von den Entscheidungen bezüglich jener Variablen ab, die mit *x* interagieren. Zur Maximierung des Performancebeitrags von Variable *x* müssen daher die Entscheidungen von allen diesen Variablen aufeinander abge-

stimmt sein. Im Fall von modulübergreifenden Interaktionen einerseits und lokal suchenden, unabhängigen Modulzulieferern andererseits, wird dieser Abstimmungsbedarf nicht adressiert und unter Umständen nicht einmal bemerkt werden. Was „gut“ für ein einzelnes Modul ist, muss aber nicht notwendigerweise auch „gut“ für das Gesamtsystem sein und umgekehrt. Es kann daher passieren, dass wirklich gute oder gar optimale Systemkonfigurationen bei der modulindividuellen Suche nie erzeugt und evaluiert werden, weil die gewählte Dekomposition und die darauf aufbauenden modulspezifischen Entwicklungspfade die Evolution des Gesamtsystems in eine andere Richtung treiben (Marengo et al. 2000, Marengo und Dosi 2005).

5 Implikationen für die verteilte Systementwicklung

Zwei verschiedene (wenngleich verwandte) Forschungsstränge erscheinen besonders hilfreich, um sich von unterschiedlichen Richtungen der Frage zu nähern, welche Implikationen sich für die verteilte Entwicklung komplexer Systeme aus den oben dargestellten theoretischen Grenzen modularer Ansätze ergeben: Zum einen die (empirische) evolutionsökonomische Literatur, die von einer Makro-Perspektive aus darauf hinweist, dass die verteilte Entwicklung in modularen Systemen nicht völlig unabhängig abläuft, sondern meist durch sogenannte Systemintegratoren („systems integrators“) gesteuert wird, d.h. mit Firmen, die von der Dekomposition des Systems, über seine modulweise Entwicklung bis zur Integration des Zielsystems eine zentrale Koordinationsrolle ausüben. Zum anderen komplexitätstheoretische Simulationsstudien zur Organisationsgestaltung, deren Mikro-Betrachtung interdependenter Entscheidungszusammenhänge zahlreiche Erkenntnisse über die Performance und die relevanten trade-offs in verteilten Problemlösungsprozessen hervorgebracht hat; die Ergebnisse lassen sich auf die verteilte Entwicklung von modularen Systemen übertragen. Im Folgenden skizzieren wir die wesentlichen Arbeiten und Erkenntnisse dieser beiden Ansätze.

5.1 Die Rolle von Systemintegratoren

Die Beschäftigung mit der Rolle von Systemintegratoren hat ihren Ursprung in der Literatur zur Innovation in komplexen Produktsystemen („complex product systems“) (Hobday 1998, Hobday et al. 2000, Davies und Hobday 2005). Komplexe Produktsysteme, etwa militärische Systeme, Telekommunikationsinfrastruktur, Flugzeuge, Satelliten- oder große Softwaresysteme, sind hochtechnologische, hochwertige Kapitalgüter, die – meist hierarchisch organisiert – aus einer Vielzahl an Subsystemen, Komponenten und Teilen bestehen (Hobday et al. 2000, Wenzel 2003). Weil jedoch die Leistungsfähigkeit solcher Systeme aus dem Zusammenspiel ihrer Einzelsysteme resultiert und diese in der Regel auf einer Vielzahl von Technologien

aufbauen und darum oft von spezialisierten Unternehmen entwickelt werden, wird eine zentrale Projektkoordination durch Systemintegratoren notwendig (Hobday et al. 2005)[5]. Systemintegration bezieht sich dabei auf die Beherrschung der Prozesse, die notwendig sind, um intern und extern entwickelte Komponenten und Wissen zu einem leistungsfähigen, komplexen System zu integrieren: „[S]ystems integration is concerned with the way in which firms and other agents bring together high-technology components, subsystems, software, skills, knowledge, engineers, managers, and technicians to produce a product in competition with other firms. The more complex, high technology, and high cost the product, the more significant systems integration becomes to the productive activity of the firm“ (Hobday et al. 2005, S. 1110).

Um dieser Aufgabe gerecht zu werden, müssen Systemhersteller Fähigkeiten auf einer Vielzahl von Gebieten besitzen (vgl. zum Folgenden etwa Hobday 1998, Prencipe 2000, Brusoni und Prencipe 2001, Brusoni et al. 2001, Prencipe et al. 2003, Ernst 2005b, Hobday et al. 2005): Zunächst müssen sie in der Lage sein, eine den jeweiligen Umständen angemessene Dekomposition eines komplexen Systems zu entwerfen, welche die intermodularen Interaktionen von Anfang bestmöglich minimiert. Ein weiteres Aufgabengebiet betrifft die verteilte Entwicklung der Module und die Koordination einer (in der Regel) Vielzahl an Modulzulieferern. Dabei kommt es insbesondere auf die genaue Spezifikation und Überwachung der Modulschnittstellen an, an denen Interaktionen auftreten können und die für das Zusammenspiel der Module im Rahmen des Gesamtsystems ausschlaggebend sind. Systemintegratoren müssen außerdem die jeweiligen Entwicklungspfade der Modulzulieferer mitverfolgen und koordinieren. Dadurch soll vermieden werden, dass die lokale Suche auf Modulebene der gewünschten globalen Optimierung auf Systemebene entgegen läuft, was dann intensive Feedback-Schleifen nach sich zöge, um das System wieder in einen ausgeglichenen Zustand zu überführen (Mihm et al. 2003). Sako (2003) führt diesbezüglich das Beispiel der Automobilindustrie an; trotz zahlreicher modularer Ansätze halten sich Automobilhersteller (original equipment manufacturers, OEMs) dort nicht etwa völlig aus den Prozessen ihrer Modulzulieferer heraus, sondern mischen sich häufig in deren Designentscheidungen ein, beispielsweise in die Wahl von Komponentenzulieferern. Ein solches „shadow engineering“ (Sako 2003, S. 239) durch den Systemhersteller ist dabei jedoch kein Ausdruck dafür, dass

[5] In der Luft- und Raumfahrt- sowie in der Verteidigungsindustrie, dem Ursprung komplexer Produktsysteme, wird ein Systemintegrator auch als „system lead“ oder „prime contractor“ bezeichnet (Sapolsky 2003).

das Konzept der Modularität falsch verstanden wäre, sondern dient letztlich der Ausübung und Aufrechterhaltung der Systemintegrationsfähigkeiten des OEM. Vor allem dort, wo die Systemeigenschaften und die Leistungsfähigkeit eines komplexen Systems nicht durch Innovationen in einem einzelnen Modul bestimmt werden, sondern sich aus dem abgestimmten Zusammenspiel vielfältiger Module ergeben, haben entsprechende koordinierende Eingriffe besondere Bedeutung (Dosi et al. 2003). Im Anschluss an die Modulentwicklung führen Systemhersteller außerdem die eigentliche Integration der Module zu einem Gesamtsystem sowie die dafür notwendigen Tests durch.

Diese vielfältigen Aufgaben können durch Methoden des Systems und Software Engineering unterstützt werden, etwa im Rahmen der Anforderungs- und Entwurfsphase, durch entsprechende detaillierte Vorgehensmodelle oder durch ein effektives Projektmanagement (Wenzel 2003, Brügge und Dutoit 2004, Davies und Hobday 2005). Erschwerend kommt jedoch hinzu, dass sich während der Modulentwicklung, also vor der Systemintegration im eigentlichen Sinne, die Leistungsfähigkeit und das Verhalten des endgültigen Systems nur schwer aus den zum jeweiligen Zeitpunkt realisierten oder geplanten Moduleigenschaften ableiten lassen. Zur Wahrnehmung ihrer Design- und Koordinationsaufgaben greifen Systemhersteller daher häufig auf Entwurfs- und Simulationssoftware zurück, um damit verschiedene Systemarchitekturen und Modulvarianten sowie das daraus resultierende Zusammenspiel in einem komplexen System im Voraus zu analysieren (Thomke und Fujimoto 2000, D'Adderio 2004). Dieser Ansatz kann ein „learning before doing“ ermöglichen, um bereits im Vorfeld unerwünschte, emergente Eigenschaften des späteren Gesamtsystem zu erkennen und sie anschließend zu vermeiden (Pisano 1996).

Systemintegratoren müssen außerdem auf eine beträchtliche Wissensbasis zugreifen können, im Gegensatz zu reinen Modullieferanten, deren Herausforderung „nur“ in der Beherrschung und Optimierung ihrer jeweiligen Modultechnologie(n) besteht. Systemhersteller benötigen dagegen einerseits „architectural knowledge“, also ein Verständnis der Systemarchitektur und damit der Zusammenhänge zwischen den Modulen und ihren jeweiligen Technologien, als Voraussetzung für eine sinnvolle Dekomposition und (im Anschluss an die Modulentwicklung) Integration zu einem leistungsfähigen Gesamtsystem (Henderson und Clark 1990). Zum anderen benötigen Systemintegratoren aber auch explizite Kenntnisse der einzelnen Modultechnologien, selbst dann, wenn die modulspezifische Wissensproduktion an Zulieferer ausgelagert wird. In diesem Zusammenhang wird auch darauf hingewiesen, dass sich das Out-

sourcing von Entwicklungstätigkeiten nicht mit dem Outsourcing von Produktionsaufgaben gleich setzen lässt (Prencipe 2000, Brusoni und Prencipe 2001, Brusoni et al. 2001). Systemhersteller „[need] to know more than they make“ (Brusoni et al. 2001), um (überraschende) Interdependenzen beherrschen und ein mögliches „System-Ungleichgewicht“ ausgleichen zu können, wie es bei unterschiedlich schnellen Entwicklungsschritten oder bei Designentscheidungen von Modulzulieferern, die suboptimal für das Gesamtsystem wären, entstehen kann. Sollten bislang unbekannte Interdependenzen auftreten, müssen Systemhersteller diese rechtzeitig erkennen und in ihrer Relevanz einschätzen können sowie angemessene Koordinierungsmaßnahmen ableiten. Ohne entsprechende Kompetenzen auch auf den ausgelagerten Technologiefeldern ließen sich diese Aufgaben nicht zufriedenstellend erfüllen. Aus diesem Grund schlägt sich auf der Ebene des Systemdesigns und der Systemintegration beim Faktor Wissen die hohe Spezialisierung und Arbeitsteilung nicht in gleicher Weise nieder wie in der eigentlichen Systemherstellung (Brusoni et al. 2001). Mit anderen Worten: Die wissensökonomische Reife von Teilaufgaben, welche aus organisationsökonomischer Sicht die Grenzen der Spezialisierung zu erklären vermag (Dietl 1993, Scheuble 1998), muss für die Metaebene des Systemführers anders definiert werden als für die Primärebene der Produktionsorganisation.

5.2 Trade-offs in verteilten Problemlösungsprozessen

In jüngerer Zeit haben verschiedene Studien damit begonnen, komplexe Systeme zu modellieren und die Implikationen verteilter Problemlösungsprozesse in Systemen anhand von Simulationsstudien zu untersuchen; auf die interessante und anspruchsvolle Methodik der Modellierungs- und Simulationsansätze, die den folgenden Ergebnisberichten zugrunde liegen, kann an dieser Stelle nur hingewiesen werden (für einen Überblick, vgl. etwa Eisenhardt und Bhatia 2002, Sorenson 2002). Arbeitsteilung in der Systementwicklung lässt sich aus abstrakter Perspektive als verteilter, adaptiver Suchprozess auffassen, bei dem den Modulentwicklern Kontrolle über eine bestimmte Untermenge der (interdependenten) Designvariablen gegeben wird, über deren Optimierung sie frei bestimmen dürfen. Diese Zusammenhänge lassen sich, wie in Abschnitt vier skizziert, modellhaft erfassen und über Simulationsrechnungen untersuchen. Ziel dieser Art von Suchbemühungen auf der Modulebene ist es, damit auch eine Vielzahl von Systemvarianten zu generieren (d.h. die Exploration auf der Systemebene zu steigern), um leistungsfähige Systemvarianten zu identifizieren und beizubehalten.

Untersuchungen weisen etwa auf einen trade-off hin, der zwischen dem Modularitätsgrad eines Systems, der Schnelligkeit der verteilten Suchprozesse sowie der Güte bzw. dem Grad der Suboptimalität der dabei erreichbaren Systemkonfigurationen besteht (Marengo et al. 2000, Marengo und Dosi 2005). Ausgangspunkt ist die Tatsache, dass ein höheres Maß an Modularität eine stärkere Reduktion des Suchraums auf Modulebene ermöglicht. Aufgrund dessen lässt sich die Systementwicklung insgesamt stärker parallelisieren und damit schneller durchführen. Auf der anderen Seite erhöht eine stärkere Modularisierung aber die Wahrscheinlichkeit, dass wichtige Interdependenzen ignoriert werden. Eine stärker modulare Systemarchitektur und die darauf aufbauende pfadabhängige und autonome Entwicklung auf Modulebene führen dann eher zum Lock-in in suboptimale Konfigurationen. Denn damit reduziert sich die Menge der generell auffindbaren Systemkonfigurationen und möglicherweise gelangen dann gerade die besonders guten Lösungen nicht ins Blickfeld. Modulare Architekturen erzwingen daher nicht selten eine trade-off-Betrachtung zwischen Flexibilität und Geschwindigkeit der verteilten Systementwicklung einerseits, und der prinzipiellen Leistungsfähigkeit der Gesamtsystems andererseits (Fleming und Sorenson 2001, Ulrich und Eppinger 2007).

Brusoni et al. (2004) weisen ebenfalls auf die Vorteile modularer Ansätze mit Blick auf die Geschwindigkeit bei der Suche nach Verbesserungen hin. Unter eher strategischen Gesichtspunkten betonen sie außerdem, dass ein höheres Maß an Modularität angesichts einer stabilen, aber komplexen Umwelt von großem Vorteil sein kann, da es eine einfache Reaktion und Veränderung auf Modulebene ermöglicht. Unter turbulenten, aber „einfachen“ Umweltbedingungen kann eine zu modulare Struktur jedoch ineffizient sein, da sie nur kleinere Änderungen auf Modulebene ermöglicht, nicht jedoch die systematischen Änderungsmaßnahmen, die angesichts einer sich rasch und tief greifend verändernden Umwelt notwendig werden können. Ethiraj und Levinthal (2004) untersuchen schließlich, welche Auswirkungen unterschiedliche (suboptimale) modulare Dekompositionen angesichts von lokaler Suche und Rekombination von Modulen auf die endgültige Systemleistung haben. Sie zeigen, dass sich im Durchschnitt eine höhere Systemperformance erreichen lässt, wenn die Systemdesigner eine – im Vergleich zu der tatsächlichen Interaktionsstruktur des Systems – zu niedrige anstelle einer zu hohen Anzahl an Modulen wählen. Durch Wahl einer kleineren Zahl an Modulen lässt sich die Gefahr von unbeachteten intermodularen Interdependenzen verringern. Nur wenn sich ein System ausschließlich aus fremden Modulen zusammensetzt und diese nicht weiter lokal angepasst werden, erweist sich eine hohes Maß an Modularität als vorteilhafter. Rekombination, so die Autoren, ist allerdings auch nur dann sinnvoll, wenn die Heterogenität der Module

hoch ist, wenn es also – wie etwa in der PC Branche – unabhängige Märkte für einzelne Module gibt, auf denen sich die „guten“ Module durchsetzen.

Bestehen jedoch intermodulare Interdependenzen und sind die Modulentwickler in ihrer Suche nach lokalen Verbesserungen völlig unabhängig, so resultiert daraus automatisch eine schlechtere Systemperformance (Ethiraj und Levinthal 2004). Siggelkow und Levinthal (2003) können jedoch zeigen, dass solche Interdependenzen nur dann schädlich sind, wenn die Idee der verteilten Suche in modularen Systemen besonders wörtlich genommen wird und die Module am Ende einfach nur „zusammengesteckt“ werden. Wenn allerdings ein Systemhersteller die verteilt entwickelten Module integriert und dabei weitere Anpassungen und Verbesserungen durchführt, kann dadurch sogar eine höhere Systemleistung erzielt werden als für den Fall, dass zuvor eine perfekte Dekomposition gefunden wurde. Der Grund liegt darin, dass unvollkommene Dekompositionen dazu führen können, dass insgesamt eine breitere Zahl an Systemvarianten generiert wird, dass also die Exploration auf der Systemebene höher ist, weil die Modulentwickler in ihrer Suche immer wieder Entscheidungen treffen, deren intermodulare Auswirkungen Veränderungen bei den jeweils anderen Modulentwickler anstoßen. Ausgehend von den dadurch erzielbaren Konfigurationen lässt sich, so das überraschende Ergebnis von Siggelkow und Levinthal, nach erfolgter Integrationsarbeit eine höhere Systemleistung erzielen. Dieser Vorteil gilt jedoch nur, wenn intermodulare Interdependenzen auch bemerkt und berücksichtigt werden, etwa durch Kommunikation zwischen den Modulherstellern (und dem Systemhersteller), und wenn der Systemhersteller über die entsprechenden Fähigkeiten zur Integration und Verbesserung der Module verfügt. Ist er dagegen ein reiner „Monteur“ der Module ohne spezifisches Modul-, Interaktions- und Integrationswissen, so führt eine anfänglich bessere Dekomposition automatisch zu einer höheren Systemperformance am Ende der Modulentwicklung im Vergleich zu einer weniger perfekten Dekomposition.

Während dadurch vor allem die Bedeutung von Systemintegratoren *nach* der eigentlichen Modulentwicklung unterstrichen wird, lässt sich im Anschluss fragen: Wie wirken sich aktive Koordinationsmaßnahmen *während* der verteilten Entwicklung aus? Rivkin und Siggelkow (2003) sowie Siggelkow und Rivkin (2006) modellieren diesbezüglich den verwandten Fall, dass ein Koordinator auf der Systemebene die Vorschläge, die bei der Exploration auf der Subsystemebene entstanden sind und an ihn weitergeleitet werden, dahingehend evaluiert, ob damit auch die Systemperformance erhöht wird. Sie können zeigen, dass eine solche koordinierende hierarchische Instanz insbesondere dann hilfreich ist, wenn die Designer auf der

Subsystemebene besonders kreativ und leistungsstark sind, wenn sie also viele verschiedene Ideen in kurzer Zeit generieren. In den genannten Fällen hilft eine aktive Hierarchie, diese breite Suche auf der Subsystemebene mit Blick auf die Systemperformance derart zu stabilisieren, dass einerseits viele unterschiedliche Lösungen erzeugt werden, andererseits aber die jeweils beste auch identifiziert und beibehalten wird. Weiterhin können sie zeigen, dass es nicht notwendigerweise schlecht und teilweise sogar vorteilhaft ist, wenn Interdependenzen zwischen den Subsystemen bestehen, solange ein aktiver Koordinator vorhanden ist, nicht jedoch, wenn die Prozesse völlig unabhängig ablaufen (Rivkin und Siggelkow 2003, Rivkin und Siggelkow 2006). Bei Interdependenzen zwischen Subsystemen haben Entscheidungen in einem Subsystem Auswirkungen auf andere Subsysteme, was dort wiederum neue Suchbemühungen anstoßen kann, etwa wenn der Koordinator die entsprechenden Entscheidungen und ihre Implikationen kommuniziert. Dadurch kann der Suchprozess länger aufrechterhalten und eine insgesamt breitere Menge an alternativen Designvarianten generiert werden, während der Koordinator darauf achtet, dass nur Varianten akzeptiert werden, welche die Leistung des Gesamtsystems steigern. Übertragen auf das Thema dieser Arbeit zeigen diese Studien damit auch, dass unvollkommene Systemdekompositionen per se keine negativen Auswirkungen haben müssen, solange die verteilte Modulentwicklung durch einen Koordinator überwacht wird, der auf den „fit“ der Module achtet.

6 Zusammenfassung und Ausblick

“The economic pressures for greater modularity … are considerable” (Pavitt 2003, S. 85), denn die in vielen Bereichen (weiter) steigende technologische Komplexität und die damit zusammenhängende Notwendigkeit zur Spezialisierung in der Wissensproduktion lassen erwarten, dass auch die Bedeutung modularer Ansätze in der Systementwicklung weiterhin zunehmen wird. Vor diesem Hintergrund hat sich dieser Artikel mit zwei Fragen beschäftigt: Zum einen mit den Vorteilen der Modularität und ihrem Beitrag zur verteilten Entwicklung komplexer Systeme, zum anderen mit der Frage, worin die Grenzen modularer Systeme bestehen und welche Implikationen sich daraus für verteilte Entwicklungsvorhaben ergeben. Die Grundaussage unseres Beitrags lässt sich dabei wie folgt zusammenfassen: Die Nutzung der „power of modularity“ (Baldwin und Clark 2000) ermöglicht ein arbeitsteiliges Vorgehen in der Systementwicklung und trägt in hohem Maße zur Komplexitätsreduktion sowie zur Verkürzung und Flexibilisierung von Entwicklungsprozessen bei. Modulare Strukturen können die Risiken komplexer (interdependenter) Systeme aber nicht völlig eliminieren, da es bei der Systemdekomposition durch begrenzt rationale Designer unvermeidbar ist, dass (teilweise

unbekannte) intermodulare Interdependenzen bestehen bleiben. Da komplexe Systeme bestenfalls auch nur „nearly decomposable“ (Simon 1996) sind, weist jede modulare Dekomposition entsprechende Schwachstellen auf. Im Rahmen der verteilten Systementwicklung durch unabhängige Modulentwickler können lokale Designentscheidungen auf der Modulebene deshalb zu unerwünschten und unvorhersehbaren Auswirkungen auf der Systemebene und damit zu einer suboptimalen Systemleistung führen, die (sofern möglich) mit hohem Integrationsaufwand behoben werden muss. Diese fundamentale Problematik führt zu einer Reihe von trade-offs, etwa zu der Beziehung, dass ein höheres Maß an Modularität zwar eine stärkere Parallelisierung und damit eine Verkürzung der Entwicklungszeit ermöglicht, zugleich aber die oben genannten negativen Effekte verstärkt (Ethiraj und Levinthal 2004, Marengo und Dosi 2005).

Derartige trade-offs zwischen den Vor- und Nachteilen modularer Systeme finden erst seit kurzem stärkere Beachtung (Marengo et al. 2000, Brusoni und Prencipe 2001, Ethiraj und Levinthal 2004, Ernst 2005b). Generell weisen diese Arbeiten darauf hin, dass eine verteilte und vollständig unabhängige Entwicklung modularer Systeme nicht möglich oder sinnvoll ist. Die modulbasierte Kodifikation von Wissen resultiert daher auf ökonomischer Ebene nicht in einer Art „Legoland“ (Pavitt 2003, S. 85), in dem sich Module, wie teilweise propagiert, aus einem Baukasten mit hoch standardisierten, austauschbaren Komponenten zu einem effizienten System kombinieren lassen (Sanchez 1995, Langlois 2002). Stattdessen müssen Systemhersteller im Rahmen der verteilten Entwicklung komplexer Systeme als zentrale Koordinatoren, also als Systemintegratoren, agieren. Die Koordinationsaufgaben von Systemintegratoren sind dabei explizit nicht nur auf die Systemdekomposition, das Outsourcing der Module und die anschließende „Systemmontage“ beschränkt, sondern beziehen sich eindeutig auch auf ein aktives Management der eigentlichen, verteilten Entwicklungsprozesse auf der Modulebene: „[M]odularity does not reduce the function of systems integration to one of simply defining architecture, subcontracting the design and production of components, and then assembling them. In complex systems, it is also important to have the competence to deal with unpredicted interactions between components, and uneven rates of development in the technologies underlying different components and subsystems” (Pavitt 2003, S. 81).

Fähigkeiten zur Systemintegration stellen somit das notwendige Gegenstück zum Outsourcing der Modulentwicklung dar: Unternehmen können die Entwicklung einzelner Module nur dann auf außen stehende Spezialisten verlagern, wenn sie die verteilte Modulentwicklung koordi-

nieren und die Module und das darin enthaltene Wissen ihrer Zulieferer anschließend zu einem leistungsfähigen Gesamtsystem integrieren können (Brusoni et al. 2001, Dosi et al. 2003). Die Betonung, dass auch modulare Systeme der Koordination bedürfen, beschränkt und erweitert die bisherige Diskussion über deren ökonomische Relevanz (Sanchez 1995, Langlois 2002): Spezialisierung und Arbeitsteilung in der Entwicklung komplexer Systeme erfordern demnach nicht nur modulare Systemarchitekturen, sondern gleichermaßen die Koordination durch Systemintegratoren, die ihr integratives Wissen an der Schnittstelle von Markt und Unternehmung einbringen und dort als "visible hand of the marketplace" zwischen Adam Smith's „invisible hand of the marketplace" und Alfred Chandler's „visible hand of the corporation" wirken (Hobday et al. 2005, S. 1136).

Ziel dieses Beitrags war es, auf die Chancen, Grenzen und Implikationen modularer Ansätze im Kontext von Komplexität und Systementwicklung hinzuweisen. Angesichts der Tatsache, dass sich die empirische und modelltheoretische Auseinandersetzung mit diesen vielschichtigen Zusammenhängen tendenziell in einem Frühstadium befindet, besteht noch breiter Forschungsbedarf. Anstatt aber spezifische Fragen aufzuwerfen, die dem Umfang und der Relevanz des Themas nur unzureichend gerecht werden könnten, skizzieren wir im Folgenden drei generelle Forschungsrichtungen, die aus unserer Sicht lohnenswertes Potential für zukünftige Studien im Spannungsfeld von Modularität und der verteilten Entwicklung komplexer Systeme bieten.

Eine Forschungsrichtung betrifft die eigentliche Koordination der verteilten Modulentwicklung durch verschiedene Koordinationsmechanismen. Arbeiten aus der Produktentwicklungsliteratur unterstreichen etwa die Bedeutung der Kommunikation im Falle von Interdependenzen zwischen einzelnen Entwicklungsaufgaben (Loch und Terwiesch 1998, Terwiesch et al. 2002). Gleichermaßen zeigen Studien zum Organisationsdesign in interdependenten Entscheidungssituationen, dass bereits einfache Hierarchien und die damit verbundenen Informationsverarbeitungs- und -filterungsprozesse einen wesentlichen Einfluss auf verteilte Explorationsbemühungen haben können (Rivkin und Siggelkow 2003, Siggelkow und Rivkin 2005, Siggelkow und Rivkin 2006). Hierarchische Strukturen sind in komplexen Entwicklungszusammenhängen in der Regel aber noch deutlicher ausgeprägt (Wenzel 2003, Hobday et al. 2005). Auf welche prinzipielle Weise wirken sich diese und andere relevante Koordinationsinstrumente auf die Beziehung zwischen der Modulebene und der daraus resultierenden Systemperformance aus? Wie lassen sich bisherige Erkenntnisse auf die Entwicklung von („near

ly decomposable“) modularen Systemen übertragen und erweitern? Und wie viel Koordinationsaufwand ist generell notwendig oder hilfreich angesichts der Tatsache, dass damit auch Kosten verbunden sind? Um konkrete Empfehlungen zur Ausgestaltung von Koordinationsmaßnahmen in modularen Entwicklungsprozessen geben zu können, wird es eine zentrale Aufgabe künftiger Forschung auf diesem Gebiet sein, die relevanten Kontingenzfaktoren und trade-offs zu identifizieren.

Eine zweite Forschungsrichtung betrifft die Affinität von Produktarchitektur und Organisationsstruktur. Einige Autoren argumentieren, dass modulare Produkte zu modularen Organisationen (und umgekehrt) führen können oder sollten (Sanchez 1995, Göpfert 1998, Takeishi 2001, Langlois 2002, MacCormack et al. 2006), wohingegen die Forschung zu Systemintegratoren gegen eine generelle 1:1-Beziehung spricht (Brusoni et al. 2001, Sosa et al. 2004, Hoetker 2006). Obwohl eine derartige Affinität zwar prinzipiell als effizient im Sinne der Arbeitsteilung gesehen wird, gibt es neben den hier diskutierten Koordinationsaspekten auch Hinweise darauf, dass sie Organisationen darin behindern kann, ihren aktuellen Fokus zugunsten neuer, leistungsfähigerer Architekturen zu ändern (Henderson und Clark 1990, Ernst 2005b). Weiterhin scheint diese Beziehung im Lebenszyklus einer Technologie unterschiedlich relevant zu sein, und nicht notwendigerweise stellt Modularität auch den „Endzustand“ einer Industrie dar, sondern häufig nur Übergang zu einer neuen, zunächst oftmals weniger modularen Technologie (Chesbrough 2003, Brusoni und Prencipe 2006, Murmann und Frenken 2006). Welche Faktoren bestimmen daher den „optimalen“ Grad an Modularität von Produkt- und Organisationssystemen und damit der arbeitsteiligen Entwicklung – auch mit Blick auf unterschiedliche Branchen (z.B. Anlagenbau, komplexe Softwaresysteme, Medien)? Wie wird diese Beziehung durch die allgemeine Wettbewerbsdynamik beeinflusst? Wie können Unternehmen in modularen Einheiten Kompetenzen entwickeln und aufrechterhalten, die über die aktuellen Module und Systeme hinausreichen und die sie für die nächste Technologiegeneration vorbereiten? Diese strategischen und dynamischen Aspekte modularer Systeme und Organisationen verdienen noch stärkere Aufmerksamkeit (Prencipe 2003, Ernst 2005b, Hoetker 2006).

Während sich die wissenschaftliche Literatur erst seit jüngerer Zeit für die prinzipiellen Möglichkeiten und Grenzen der Modularität interessiert, zeigen vereinzelte empirische Studien realer Produktsysteme, dass die Verwendung modularer Ansätze in der Realität meist weitaus „schmutziger“ verläuft als in der Theorie (MacCormack et al. 2006). Andere, weniger (gute)

modulare Produktarchitekturen können oft die gleiche Leistung erzielen wie gute modulare Dekompositionen. In realen Systemen lassen sich teilweise sogar intermodulare Interdependenzen reduzieren oder die Zahl der Module ändern, ohne dass dies Auswirkungen auf die Systemperformance hätte (MacCormack et al. 2006). Es darf daher vermutet werden, dass viele modulare Systemarchitekturen in der Praxis oftmals noch um einiges von der eigentlichen „performance frontier“ entfernt sind (MacCormack et al. 2006, S. 25) und ihr theoretisches Potential, auch mit Blick auf die Arbeitsteilung in der Systementwicklung, damit noch nicht vollständig ausgeschöpft ist. Wie hat sich diese Vielfalt an gleichermaßen leistungsfähigen Architekturen entwickelt und ist sie sinnvoll (und wenn ja, warum)? Wann und wie kann dieser Spielraum verringert werden, um ein modulares System näher an die „performance frontier“ zu führen und ein höheres Maß an Arbeitsteilung zu ermöglichen? Wann führt eine stärkere Modularisierung stattdessen zu trade-offs mit anderen Designelementen? Diese Frage stellt sich etwa, aber nicht nur, im Rahmen von proprietär entwickelter Software, die zur Weiterentwicklung in die Open Source Welt überführt werden soll. Trotz einer Vielzahl theoretischer Argumente sind die empirischen Erkenntnisse zu diesen Fragen noch sehr gering, insbesondere auf der Ebene konkreter modularer Systeme. Diese Lücke zu schließen ist Anliegen der dritten von uns vorgeschlagenen weiteren Forschungsrichtung.

Literatur

Alexander, C (1964) Notes on the Synthesis of Form. Harvard University Press, Cambridge, MA

Baldwin, CY, Clark, KB (2000) Design Rules: The Power of Modularity. MIT Press, Cambridge, MA

Baldwin, CY, Clark, KB (2006) The architecture of participation: does code architecture mitigate free riding in the open source development model? Manage Sci 52:1116-1127

Bar-Yam, Y (1997) Dynamics of Complex Systems. Addison-Wesley, Reading, MA

Booch, G, Maksimchuk, RA, Engel, MW, Young, BJ, Conallen, J, Houston, KA (2007) Object-Oriented Analysis and Design with Applications. Addison-Wesley, Upper Saddle River, NJ

Brügge, B, Dutoit, AH (2004) Object-Oriented Software Engineering: Conquering Complex and Changing Systems. Prentice Hall, Upper Saddle River, NJ

Brügge, B, Harhoff, D, Picot, A, Creighton, O, Fiedler, M, Henkel, J (2004) Open-Source-Software: Eine ökonomische und technische Analyse. Springer, Berlin

Brusoni, S, Marengo, L, Prencipe, A, Valente, M (2004) The value and costs of modularity: a cognitive perspective. Europ Manage Rev 4:121-132

Brusoni, S, Prencipe, A (2001) Unpacking the black box of modularity: technologies, products and organizations. Ind Corp Change 10:179-205

Brusoni, S, Prencipe, A (2006) Making design rules: a multidomain perspective. Organ Sci 17:179-189

Brusoni, S, Prencipe, A, Pavitt, K (2001) Knowledge specialization, organizational coupling, and the boundaries of the firm: why do firms know more than they make? Admin Sci Quart 46:597-621

Burton, RM, Obel, B (1998) Strategic Organizational Diagnosis and Design: Developing Theory for Application. Kluwer, Dordrecht

Chapman, WL, Rozenblit, J, Bahill, AT (2001) System design is an NP-complete problem. Systems Engineering 4:222-229

Chesbrough, H (2003) Towards a Dynamics of Modularity: A Cyclical Model of Technical Advance. In: Prencipe, A, Davies, A, Hobday, M (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford, 174-198

Cohen, WM, Levinthal, DA (1989) Innovation and learning: the two faces of R&D. Econ J 99:569-595

Cohen, WM, Levinthal, DA (1990) Absorptive capacity: a new perspective on learning and innovation. Admin Sci Quart 35:128-152

Cyert, RM, March, JG (1963) A Behavioral Theory of the Firm. Prentice Hall, Englewood Cliffs, NJ

D'Adderio, L (2004) Inside the Virtual Product: How Organizations Create Knowledge through Software. Edward Elgar, Cheltenham

Davies, A, Hobday, M (2005) The Business of Projects: Managing Innovation in Complex Products and Systems. Cambridge University Press, Cambridge

Dietl, H (1993) Institutionen und Zeit. Mohr, Tübingen

Dosi, G, Hobday, M, Marengo, L, Prencipe, A (2003) The Economics of Systems Integration: Towards an Evolutionary Interpretation. In: Hobday, M, Prencipe, A, Davies, A (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford, 95-113

Eisenhardt, KM, Bhatia, MM (2002) Organizational Complexity and Computation. In: Baum, JAC (Hrsg) Companion to Organizations. Blackwell Publishers, Oxford, 442-466

Eppinger, SD, Chitkara, AR (2006) The new practice of global product development. MIT Sloan Manage Rev 49:22-30

Eppinger, SD, Whitney, DE, Smith, R, Gebala, D (1994) A model-based method for organizing tasks in product development. Res Eng Des 6:1-13
Erl, T (2008) SOA Principles of Service Design. Prentice Hall, Upper Saddle River, NJ
Ernst, D (2005a) Internationalisation of innovation: why is chip design moving to Asia? Int J Innov Manage 9:47-73
Ernst, D (2005b) Limits to modularity: reflections on recent developments in chip design. Industry and Innovation 12:303-335
Ernst, D, Kim, L (2002) Global production networks, knowledge diffusion, and local capability formation. Res Policy 31:1417-1429
Ethiraj, SK, Levinthal, D (2004) Modularity and innovation in complex systems. Manage Sci 50:159-173
Fleming, L, Sorenson, O (2001) Technology as a complex adaptive system: evidence from patent data. Res Policy 30:1019-1039
Fleming, L, Sorenson, O (2003) Navigating the technology landscape of innovation. MIT Sloan Manage Rev 44:15-23
Galbraith, J (1973) Designing Complex Organizations. Addison-Wesley, Reading, MA
Gavetti, G, Levinthal, DA (2000) Looking forward and looking backward: cognitive and experiential search. Admin Sci Quart 45:113-137
Göpfert, J (1998) Modulare Produktentwicklung: Zur gemeinsamen Gestaltung von Technik und Organisation. Gabler, Wiesbaden
Hagedoorn, J (2002) Inter-firm R&D partnerships: an overview of major trends and patterns since 1960. Res Policy 31:477-492
Hagel, J, Brown, JS (2001) Your next IT strategy. Harvard Bus Rev 79:105-113
Henderson, R, Clark, KB (1990) Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms. Admin Sci Quart 35:9-30
Hobday, M (1998) Product complexity, innovation and industrial organisation. Res Policy 26:689-710
Hobday, M, Davies, A, Prencipe, A (2005) Systems integration: a core capability of the modern corporation. Ind Corp Change 14:1109-1143
Hobday, M, Rush, H, Tidd, J (2000) Innovation in complex products and systems. Res Policy 29:793-804
Hoetker, G (2006) Do modular products lead to modular organizations? Strategic Manage J 27:501-518
Hoetker, G, Swaminathan, A, Mitchell, W (2007) Modularity and the impact of buyer-supplier relationships on the survival of suppliers. Manage Sci 53:178-191
Holland, JH, Holyoak, KJ, Nisbett, RE, Thagard, PR (1986) Induction: Processes of Inference, Learning, and Discovery. MIT Press, Cambridge, MA
Kauffman, S (1993) Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York
Kauffman, S (1995) At Home in the Universe: The Search for the Laws of Self-Organization and Complexity. Oxford University Press, New York
Khandwalla, PN (1977) The Design of Organizations. Harcourt Brace Jovanovich, New York
Langlois, RN (2002) Modularity in technology and organization. J Econ Behav Organ 49:19-37
Langlois, RN (2003) The vanishing hand: the changing dynamics of industrial capitalism. Ind Corp Change 12:351-385
Levinthal, DA (1997) Adaptation on rugged landscapes. Manage Sci 43:934-950
Levinthal, DA, March, JG (1982) A model of adaptive organizational search. J Econ Behav Organ 2:307-333
Levinthal, DA, March, JG (1993) The myopia of learning. Strategic Manage J 14:95-112

Loch, CH, Terwiesch, C (1998) Communication and uncertainty in concurrent engineering. Manage Sci 44:1032-1048
Loch, CH, Terwiesch, C, Thomke, S (2001) Parallel and sequential testing of design alternatives. Manage Sci 47:663-678
Löwer, UM (2006) Interorganisational Standards: Managing Web Services Specifications for Flexible Supply Chains. Physica, Heidelberg
MacCormack, A, Rusnak, J, Baldwin, CY (2006) Exploring the structure of complex software designs: an empirical study of open source and proprietary code. Management Science 52:1015-1030
March, JG (1991) Exploration and exploitation in organizational learning. Organ Sci 2:71-87
March, JG, Simon, HA (1958) Organizations. Wiley, New York
Marengo, L, Dosi, G (2005) Division of labor, organizational coordination and market mechanism in collective problem-solving. J Econ Behav Organ 58:303-326
Marengo, L, Dosi, G, Legrenzi, P, Pasquali, C (2000) The structure of problem-solving knowledge and the structure of organizations. Ind Corp Change 9:757-788
Mihm, J, Loch, C, Huchzermeier, A (2003) Problem-solving oscillations in complex engineering projects. Manage Sci 49:733-750
Murmann, JP, Frenken, K (2006) Toward a systematic framework for research on dominant designs, technological innovations, and industrial change. Res Policy 35:925-952
Nelson, RR, Winter, SG (1982) An Evolutionary Theory of Economic Change. Harvard University Press, Cambridge, MA
Newell, A, Simon, HA (1972) Human Problem Solving. Prentice Hall, Englewood Cliffs, NJ
O'Reilly, T (1999) Lessons from open source software development. Commun ACM 42:33-37
Orton, JD, Weick, KE (1990) Loosely coupled systems: a reconceptualization. Acad Manage Rev 15:203-223
Parnas, DL (1972) On the criteria to be used in decomposing systems into modules. Commun ACM 15:1053-1058
Pavitt, K (2003) Specialization and Systems Integration: Where Manufacture and Services Still Meet. In: Hobday, M, Prencipe, A, Davies, A (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford, 78-91
Picot, A, Baumann, O (2007) Die verteilte Entwicklung komplexer Systeme: Grenzen der Modularität und Fähigkeiten zur Systemintegration. In: Blum, U, Eckstein, A, Eckstein, A (Hrsg) Wirtschaftsinformatik im Fokus der modernen Wissensökonomik - Festschrift für Professor Dr. Dr. h.c. Wolfgang Uhr. TUDpress, Dresden, 335-352
Picot, A, Freudenberg, H (1998) Neue organisatorische Ansätze im Umgang mit Komplexität. In: Adam, D (Hrsg) Komplexitätsmanagement. Gabler, Wiesbaden, 69-86
Picot, A, Reichwald, R, Wigand, RT (2003) Die grenzenlose Unternehmung. Gabler, Wiesbaden
Picot, A, Ripperger, T, Wolff, B (1996) The fading boundaries of the firm: the role of information and communication technology. J Inst Theor Econ 152:65-79
Pisano, GP (1990) The R&D boundaries of the firm: an empirical analysis. Admin Sci Quart 35:153-176
Pisano, GP (1996) Learning-before-doing in the development of new process technology. Res Policy 25:1097-1119
Powell, WW, Koput, KW, Smith-Doerr, L (1996) Interorganizational collaboration and the locus of innovation: networks of learning in biotechnology. Admin Sci Quart 41:116-145
Prencipe, A (2000) Breadth and depth of technological capabilities in CoPS: the case of the aircraft engine control system. Res Policy 29:895-911

Prencipe, A (2003) Corporate Strategy and Systems Integration Capabilities: Managing Networks in Complex Systems Industries. In: Prencipe, A, Davies, A, Hobday, M (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford,
Prencipe, A, Davies, A, Hobday, M (2003) (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford.
Rivkin, JW, Siggelkow, N (2003) Balancing search and stability: interdependencies among elements of organizational design. Manage Sci 49:290-311
Rivkin, JW, Siggelkow, N (2006) Organizing to strategize in the face of interactions: preventing premature lock-in. Long Rang Plann 39:591-614
Sako, M (2003) Modularity and Outsourcing: The Nature of Co-evolution of Product Architecture and Organization Architecture in the Global Automotive Industry. In: Prencipe, A, Davies, A, Hobday, M (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford, 229-253
Sanchez, R (1995) Strategic flexibility in product competition. Strategic Manage J 16:135-159
Sapolsky, HM (2003) Inventing Systems Integration. In: Prencipe, A, Davies, A, Hobday, M (Hrsg) The Business of Systems Integration. Oxford University Press, Oxford, 15-34
Schaefer, S (1999) Product design partitions with complementary components. J Econ Behav Organ 38:311-330
Scheuble, S (1998) Wissen und Wissenssurrogate: Eine Theorie der Unternehmung. Gabler, Wiesbaden
Schilling, MA (2000) Toward a general modular systems theory and its application to interfirm product modularity. Acad Manage Rev 25:312-334
Schilling, MA, Steensma, HK (2001) The use of modular organizational forms: an industry-level analysis. Acad Manage J 44:1149-1168
Schumpeter, JA (1939) Business Cycles. McGraw-Hill, New York
Siggelkow, N, Levinthal, DA (2003) Temporarily divide to conquer: centralized, decentralized, and reintegrated organizational approaches to exploration and adaptation. Organ Sci 14:650-669
Siggelkow, N, Rivkin, JW (2005) Speed and search: designing organizations for turbulence and complexity. Organ Sci 16:101-122
Siggelkow, N, Rivkin, JW (2006) When exploration backfires: unintended consequences of multilevel organizational search. Acad Manage J 49:779-795
Simon, HA (1955) A behavioral model of rational choice. Q J Econ 69:99-118
Simon, HA (1956) Rational choice and the structure of the environment. Psychol Rev 63:129-138
Simon, HA (1962) The architecture of complexity. Proceedings of the American Philosophical Society 106:467-482
Simon, HA (1991) Organizations and markets. J Econ Perspect 5:25-44
Simon, HA (1996) The Sciences of the Artificial. MIT Press, Cambridge, MA
Simon, HA (2002) Near decomposability and the speed of evolution. Ind Corp Change 11:587-599
Sorenson, O (2002) Interorganizational Complexity and Computation. In: Baum, JAC (Hrsg) Companion to Organizations. Blackwell Publishers, Oxford, 664-685
Sosa, ME, Eppinger, SD, Rowles, CM (2004) The misalignment of product architecture and organizational structure in complex product development. Manage Sci 50:1674-1689
Staudenmeyer, N, Tripsas, M, Tucci, CL (2005) Interfirm modularity and its implications for product development. J Prod Innovat Manag 22:303-321
Steward, DV (1981) The design structure system: a method for managing the design of complex systems. IEEE T Eng Manage 28:71-74

Stuart, TE, Podolny, JM (1996) Local search and the evolution of technological capabilities. Strategic Manage J 17:21-38
Takeishi, A (2001) Bridging inter- and intra-firm boundaries: management of supplier involvement in automobile product development. Strategic Manage J 22:403-433
Terwiesch, C, Loch, CH, De Meyer, A (2002) Exchanging preliminary information in concurrent engineering: alternative coordination strategies. Organ Sci 13:402-419
Thomke, S, Fujimoto, T (2000) The effect of "front-loading" problem-solving on product development performance. J Prod Innovat Manag 17:128-142
Thompson, JD (1967) Organizations in Action. McGraw-Hill, New York
Tversky, A, Kahneman, D (1986) Rational choice and the framing of decisions. J Bus 59:251-278
Ulrich, KT (1995) The role of product architecture in the manufacturing firm. Res Policy 24:419-440
Ulrich, KT, Eppinger, SD (2007) Product Design and Development. McGraw-Hill, New York
Veryard, R (2001) The Component-Based Business: Plug and Play. Springer, London
von Hippel, E (1990) Task partitioning: an innovation process variable. Res Policy 19:407-418
von Hippel, E (1994) "Sticky information" and the locus of problem solving: implications for innovations. Manage Sci 40:429-439
Weick, KE (1976) Educational organizations as loosely coupled systems. Admin Sci Quart 21:1-19
Wenzel, S (2003) Organisation und Methodenauswahl in der Produktentwicklung. Herbert Utz, München
Williamson, OE (1975) Markets and Hierarchies: Analysis and Antitrust Implications: A Study in the Economics of Internal Organization. Free Press, New York