

Web 2.0 und SOA – verwandte Konzepte?

Eine neue Klasse von Internetapplikationen wie interaktive Enzyklopädien, Blogs oder auch sogenannte Mashups hat während der letzten Jahre große Verbreitung gefunden und wird oft unter dem Begriff Web 2.0 zusammengefasst. Ein Thema, das seit kurzer Zeit diskutiert wird, ist der Zusammenhang zwischen Web 2.0 und dem Konzept der serviceorientierten Architektur (SOA). Der Gedanke der Wiederverwendbarkeit und des Verbergens von Komplexität, aber auch das lose Koppeln verschiedener Dienste hat die wissenschaftliche Gemeinschaft angeregt, Gemeinsamkeiten zwischen Web 2.0 und SOA herauszuarbeiten. In dieser Arbeit werden Anwendungsbeispiele der beiden Konzepte sowohl aus technischer als auch aus betriebswirtschaftlicher Perspektive untersucht und auf der Basis von sieben Kriterien kontrastiert. Die detaillierte Analyse soll dabei Unterschiede wie auch Gemeinsamkeiten aufdecken und schließlich einen Ausblick auf zukünftige Applikationen, die auf Technologien und Gestaltungsprinzipien von sowohl Web 2.0 als auch SOA aufbauen, geben.

Inhaltsübersicht

- 1 Einleitung und Begriffsdefinitionen
- 2 Web 2.0 und SOA – eine detaillierte Betrachtung
 - 2.1 Gestaltung des Produkts
 - 2.2 Technische Basis
 - 2.3 Potenzielle Kunden
 - 2.4 Wertschöpfungsketten
 - 2.5 Erlösmodelle
 - 2.6 Warenfluss
 - 2.7 Einflüsse des sozialen Umfelds
- 3 Konvergenz der beiden Konzepte: Das Internet der Dienste
- 4 Schlussbemerkungen
- 5 Literatur

1 Einleitung und Begriffsdefinitionen

Der Begriff Web 2.0 wurde ursprünglich von [O'Reilly 2007] geprägt und beschreibt ein neuartiges Konzept, das einer wachsenden Anzahl von internetbasierten Applikationen wie Blogs, interaktiven Enzyklopädien sowie Mashups innewohnt. Das Internet wird dabei als Plattform betrachtet, die das Erstellen von Anwendungen oder Inhalten erlaubt, deren Besitzer nicht notwendigerweise klar definiert sind. Sie werden vielmehr aus verschiedenen anderen, meist webbasierten Ressourcen mithilfe offener Standards und Protokollen zu Anwendungen zusammengefügt. Die Nutzung der sogenannten kollektiven Intelligenz und der Einsatz einfacher Programmierkonzepte stellen weitere zentrale Charakteristika dar. Da die meisten der Web 2.0-Applikationen nicht als statische Artefakte im Handel bezogen werden, sondern als webbasierte Dienste verfügbar sind, proklamiert O'Reilly auch das Ende des Software-Releasezyklus. Web 2.0-Anwendungen sind darüber hinaus oft durch intuitive und teilweise Desktop-ähnliche Benutzerschnittstellen gekennzeichnet.

Die serviceorientierte Architektur (SOA) hat in den letzten Jahren als IT-Designkonzept an Bedeutung gewonnen: Applikationen werden als Services mit einheitlich beschriebenen Schnittstellen gekapselt und über bestimmte Discovery-Services frei zugänglich gemacht [Alonso et al. 2004]. Servicekonsumenten können so Services beliebiger Anbieter finden und diese mit eigenen Applikationen koppeln. Die Serviceschnittstellen verbergen dabei die der gekapselten Funktionalität inhärente Komplexität. Sie bieten nach außen hin lediglich eine einfache Beschreibung der unterstützten Dienste und der Nachrichten, die vom Service

empfangen oder gesendet werden können. Ein wichtiger Vorteil von SOA gegenüber herkömmlichen Prinzipien des Softwaredesigns stellt die Möglichkeit dar, Services mit relativ geringem Programmieraufwand schnell über standardisierte Schnittstellen zu koppeln oder sie zum Aufbau neuer, komplexerer Dienste oder Anwendungen zu komponieren.

Aufgrund einiger Ähnlichkeiten von Web 2.0- und SOA-Anwendungen scheint eine vergleichende, detaillierte Analyse sowohl auf der Grundlage von technischen als auch ökonomische Gesichtspunkten hoch relevant [McAfee 2007]. Die sieben Dimensionen des MCM Business Model Framework [Högg et al. 2006], die Ansätze des Geschäftsmodell-Rahmenwerkes von [Timmers 1998] verwenden, werden als Grundlage hierfür genutzt. Ziel ist es, eine fundierte Analyse in die unter Experten entbrannte Diskussion einzubringen, Gemeinsamkeiten und Unterschiede herauszuarbeiten und anhand von Fallbeispielen Trends zu identifizieren, die sich aus einer möglichen Konvergenz von Designprinzipien und Technologien aus den Bereichen Web 2.0 und SOA ergeben können. Diese Trends dienen schließlich dazu, Designvorschläge und Handlungsempfehlungen für zukünftige Anwendungen im Unternehmensumfeld ableiten zu können.

2 Web 2.0 und SOA – eine detaillierte Betrachtung

2.1 Gestaltung des Produkts

Die Analyse von populären Web 2.0-Applikationen [Högg et al. 2006] hat folgende Aufteilung in drei Klassen ergeben:

- *Communities*
Diese bieten Nutzern die Möglichkeit, einem gemeinsamen Interesse nachzugehen (Netzwerkplattformen wie LinkedIn.com) oder Wissen zu teilen (Swicki.eurester.com).
- *Plattformen*
Sie ermöglichen es Nutzern, Inhalte oder auch Applikationslogik selbst zu erstellen

und mit einem breiten Publikum zu teilen. Blogs und Onlineverzeichnisse, die Navigationsdienste und von Nutzern generierte Taxonomien anbieten, aber auch Mashup-Plattformen stellen Beispiele für diese Kategorie dar. Der Begriff Mashup wird weithin benutzt, um die freie, von Nutzern bestimmte Vernetzung und Komposition verschiedener Ressourcen im Internet zu beschreiben. *Yahoo! Pipes* (pipes.yahoo.com) beispielsweise erlaubt Nutzern, ohne Programmieraufwand Informationen (RSS-basiert) von beliebigen Quellen heranzuziehen, sie zu filtern, zu vermischen und wieder anzubieten.

- *Tools zur Online-Zusammenarbeit*
Diese Tools stellen die dritte Klasse von Web 2.0-Anwendungen dar und schließen Beispiele wie kollaborative Textverarbeitung oder Online-Brainstorming ein.

Die Analyse von SOA-Anwendungen, die sich aus Fallbeispielen von großen Softwareanbietern wie SAP und IBM sowie Analysten wie Gartner konstituieren, hat folgende Klassifizierung ergeben:

- *Organisationsübergreifende SOA-Anwendungen*
Diese ermöglichen es Unternehmen, Geschäftstransaktionen auf effiziente und flexible Weise zu automatisieren [McAfee 2005]. Ihr Einsatz steigert die Effizienz der Zusammenarbeit zwischen Unternehmen und fördert die Entstehung virtueller Unternehmen, ist aber aufgrund der technischen Komplexität und des Mangels an Standards in der Praxis noch wenig verbreitet.
- *Organisationsinterne Integration heterogener Systeme*
Firmen, die zum Beispiel als Folge von Übernahmen disparate Applikationen zu harmonisieren versuchen, können mithilfe von SOA eine konsistente und gleichzeitig schnell modifizierbare IT-Landschaft realisieren. Applikationen werden einheitlich gekapselt und können dann je nach Anforderung ge-

koppelt oder zu neuen Services komponiert werden.

- *Prinzip bei der Entwicklung von Applikationen*
Anstatt Funktionalität immer neu zu programmieren, wird mithilfe von SOA-basierten Entwicklungsumgebungen Programmiercode mit bestimmter Funktionalität als wiederverwendbarer Service gekapselt.

Eine wichtige Gemeinsamkeit zwischen den Anwendungen der beiden Konzepte Web 2.0 und SOA besteht in der Kapselung von Funktionalität in separaten Services und der Möglichkeit, diese den aktuellen Anforderungen entsprechend zu komponieren. Web 2.0-Anwendungen erlauben ihren Nutzern, Inhalte und Funktionalität zu nutzen und mit anderen Web-Ressourcen zu vermischen. SOA-Anwendungen folgen dem gleichen Ansatz und ermöglichen es Firmen, als Services gekapselte Applikationen wiederzuverwenden und beliebig zu komponieren. Daraus leitet sich eine weitere gemeinsame Eigenschaft beider Konzepte ab: die Agilität. Sowohl Web 2.0- als auch SOA-Anwendungen können von ihren Nutzern mit relativ geringem Aufwand an aktuelle Anforderungen angepasst werden. Gerade die oben aufgeführten Web 2.0-Plattformen beziehen einen großen Teil ihres Mehrwerts aus der Eigenschaft, dass auf ihnen angebotener Inhalt oder bereitgestellte Funktionalität sich ständig an die Bedürfnisse der Nutzer anpassen lässt. Das sogenannte »User Empowerment« repräsentiert eine weitere, entscheidende Gemeinsamkeit von Produkten aus den beiden Bereichen. Applikationen sind nicht mehr monolithische, statische Blöcke, sondern sollen von ihren jeweiligen Nutzern und nicht mehr ausschließlich von Entwicklern angepasst werden können.

Ein wesentlicher Unterschied zwischen beiden Prinzipien ist die soziale Komponente. SOA-Anwendungen werden oft zur Kopplung disparater Geschäftsfunktionalitäten etabliert, aber fördern nur indirekt menschliche Zusammenarbeit. Web 2.0-Anwendungen hingegen sind

in vielen Fällen der Kollaboration und der Kommunikation von Menschen mithilfe von Technologie gewidmet. Im Gegensatz zu SOA-Applikationen gehen sie außerdem meist mit einem intuitiven und benutzerfreundlichen Nutzer-Interface einher. Sie verstehen den Menschen explizit als essenziellen Bestandteil des Wertbeitrags und binden ihn deshalb über ansprechende Interfaces ein. Der dritte wesentliche Unterschied wurde in dem Grad der Ex-ante-Bestimmung und der Involvierung von Kontrollprozessen der Produkte festgestellt. Web 2.0-Anwendungen unterliegen in den seltensten Fällen strukturellen oder inhaltlichen Restriktionen und entwickeln sich frei den aktuellen Bedürfnissen ihrer Nutzer entsprechend. SOA-Anwendungen werden dagegen oft im professionellen Umfeld eingesetzt und unterliegen strikten Governance-Prozessen [Weill & Ross 2004], um ein nicht mehr entwirrbares Netz von Serviceverbindungen und Nachrichtenmustern zu vermeiden und den fehlerfreien Betrieb von Unternehmens-IT zu gewährleisten.

2.2 Technische Basis

Die hohe Verbreitung von Web 2.0-Anwendungen geht nicht mit neuen fundamentalen technologischen Innovationen einher, ist aber durch eine neuartige Nutzung schon bestehender und einiger neuer Standards und Protokolle gekennzeichnet. Zu diesen Standards zählen unter anderen XML, RSS, REST und Ajax.

- *XML*
XML erfreut sich großer Akzeptanz und wird zur strukturierten Modellierung und Beschreibung von Daten eingesetzt.
- *RSS*
RSS beschreibt ein XML-Format mit sehr wenigen Markups und erlaubt die einfache und effiziente Aggregation von Informationen aus verschiedenen Quellen.
- *REST*
REST (Representational State Transfer) ermöglicht die einfache und einheitliche Adres-

sierung von internetbasierten Ressourcen und erlaubt es Nutzern, mithilfe von XML-Nachrichten bestimmten Formats kleine Applikationen zu koppeln [Fielding 2000].

- **Ajax**
Ajax (Asynchronous JavaScript and XML) wird benutzt, um reichhaltige und intuitive Web-Benutzeroberflächen zu erstellen, die Desktop-ähnliche Funktionalitäten besitzen.

SOA-Anwendungen können mithilfe von verschiedensten Technologien realisiert werden, jedoch sind WSDL- und SOAP-basierte Web-Services als technische Grundlage sehr weit verbreitet [Alonso et al. 2004]. Im Kern besteht eine serviceorientierte Architektur aus Anbietern von Services, die mithilfe von WSDL beschrieben werden, einem oder mehreren Serviceregistern (z. B. auf Basis von UDDI) und den Servicekonsumenten. Konsumenten finden die Services über diese Register und verbinden sich daraufhin. Im Laufe der letzten Jahre hat sich ein umfangreicher Stack diverser Standards entwickelt, der Web-Service-basierte SOAs unterstützt (Abb. 1). Transportprotokolle wie HTTP bilden die Basis dieses Stacks, während Daten, die zwischen den Services ausgetauscht werden, oftmals in XML beschrieben werden. WSDL ermöglicht die einheitliche Definition von Serviceschnittstellen.

UDDI spezifiziert Mechanismen, um Services über Register leicht auffindbar zu machen, während SOAP das Nachrichtenformat und ein Austauschprotokoll zwischen Services definiert. Wenig verbreitete Standards für die Semantik von ausgetauschten Informationen stellen oft große Hindernisse für die Interoperabilität verschiedener Services dar. BPEL wird vielfach für die Orchestrierung und Koordination einzelner Services, also zum Beispiel zur ausführbaren Definition eines übergreifenden elektronischen Geschäftsprozesses, verwendet.

Beide Konzepte ermöglichen die Kopplung verteilter Applikationen über einheitliche Schnittstellen, jedoch unterscheiden sich die damit verbundenen Protokolle und Standards insbesondere in Bezug auf die inhärente Komplexität. Zudem zielen die Technologien aus beiden Bereichen auf das Verbergen interner Komplexität und die Reduzierung von Programmieraufwand ab. Standards wie Ajax unterstützen nicht nur die Entwicklung von benutzerfreundlichen Web-Benutzeroberflächen, sondern erlauben es selbst wenig technikaffinen Menschen, neue Applikationen oder Inhalte per Drag & Drop zusammenzustellen. Nutzer ändern so ihre Rolle von reinen Konsumenten hin zu Mitgestaltern der nachgefragten An-

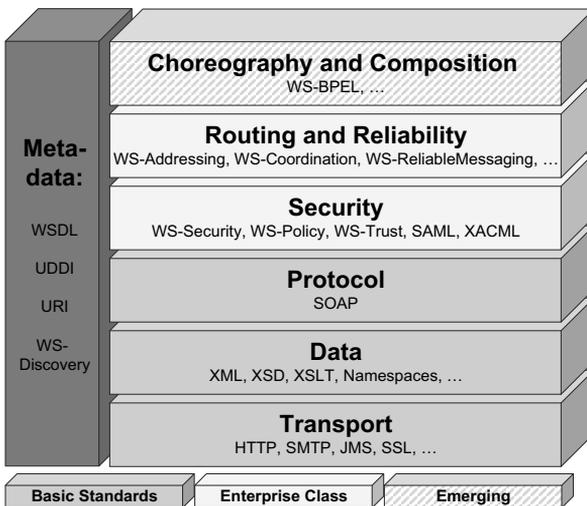


Abb. 1: Übersicht über einige Standards des Web-Service-Stacks

wendung. SOA-Anwendungen folgen im Unternehmensumfeld dem gleichen Ansatz: Existierende Services können mit geringem Aufwand wiederverwendet und neu komponiert werden.

Ein erster wesentlicher Unterschied betrifft die semantische Interoperabilität. Web 2.0-Anwendungen komponieren Inhalte oder Funktionalität mit dem Ziel, Menschen direkt über Benutzer-Interfaces zu bedienen. SOA-Anwendungen hingegen sind begrenzt auf die Interaktion von Maschinen, die auf mögliche Fehler intolerant reagieren. Designer müssen daher die gleiche semantische Interpretation aller Daten durch alle in einer Anwendung involvierten Services sicherstellen, um Fehler bei der Ausführung zu vermeiden. Über Web 2.0-Plattformen aggregierte Inhalte sind dagegen oft fehlerhaft, werden aber aufgrund von Flexibilität und der menschlichen Fähigkeit zur Fehlerkorrektur dennoch meist richtig verstanden, was die Einhaltung starrer Standards relativiert und somit das schnelle Wachstum der betreffenden Anwendungen fördert. Ein zweiter wichtiger Punkt betrifft die funktionalen Restriktionen auf technischer Ebene. Standards aus dem Web 2.0-Bereich (zum Beispiel RSS) stellen dem Benutzer oftmals nur sehr wenige Optionen zur Verfügung, was die Anzahl verschiedener Implementierungen reduziert, jedoch die Interoperabilität und letztlich die Verbreitung der entsprechenden Anwendungen steigert. Standards aus dem Bereich SOA können als komplexer und ausgereifter bezeichnet werden, weisen allerdings auch deutlich mehr Freiheitsgrade auf und erschweren dadurch die nahtlose Interoperabilität von Services. Ein dritter Unterschied betrifft die Art der Kopplung von Ressourcen. Während Web 2.0 heute meist auf Syndizierung, also die reine Zusammenführung von Inhalten, fokussiert, werden bei SOA-Anwendungen Services koordiniert. Koordination bezeichnet in diesem Kontext den Einsatz von komplexen Protokollen, die verschiedene Services in einer bestimmten zeitlichen Abfolge

aufrufen, um einen Gesamtprozess abzubilden (Choreographie).

2.3 Potenzielle Kunden

Potenzielle Kunden von Produkten oder Dienstleistungen im Web 2.0-Umfeld umfassen prinzipiell jeden Internetnutzer [Högg et al. 2006]. Web 2.0-Anwendungen zielen allgemein auf den »Long Tail« ab [Anderson 2006], also die breite Masse einzelner Nutzer oder kleiner Firmen, und fokussieren weniger auf einzelne große Benutzer. Diese Ausrichtung kommt dem grundlegenden Prinzip der Maximierung kollektiver Intelligenz zugute: Erst durch eine signifikante Zahl aktiver Nutzer, die als Mitgestalter agieren, erlangen viele Produkte einen hohen Mehrwert. Die Kundenbedürfnisse, die durch Web 2.0-Anwendungen bedient werden, lassen sich sehr gut entlang der drei obersten Ebenen von Maslows Bedürfnispyramide strukturieren. Nutzer können sich demnach durch aktive Partizipation selbst verwirklichen, durch Kommentare anderer oder durch den Grad der Weiterverbreitung der eigenen Beiträge soziale Anerkennung erlangen und schließlich auch soziale Beziehungen initiieren und pflegen. Im Rahmen der Analysen zahlreicher Anwendungsfälle haben sich vor allem mittlere und größere Unternehmen als hauptsächliche Kundengruppe von SOA-Produkten herausgestellt. Sie bezwecken, interne Applikationslandschaften zu harmonisieren oder effiziente und leicht modifizierbare elektronische Geschäftsbeziehungen mit anderen Firmen aufzusetzen. Insofern stehen hier im Gegensatz zu Web 2.0 weniger menschliche Bedürfnisse, sondern eher wirtschaftliche Interessen von Firmen im Vordergrund.

2.4 Wertschöpfungsketten

Im Web 2.0-Kontext werden traditionelle Wertschöpfungsketten aufgebrochen und durch lose gekoppelte Netzwerke von Anbietern und Konsumenten ersetzt. Aufgrund der Einfachheit und freien Verfügbarkeit von Protokollen

und Standards können auch Internetnutzer mit geringer IT-Affinität Inhalte oder Funktionalitäten erstellen, publizieren und mit anderen komponieren. Im Gegensatz zu diesem »m:n«-Modell der Wertschöpfungsketten weisen Wertschöpfungsketten im SOA-Umfeld meist einen »Experten«, wie zum Beispiel SAP oder IBM, auf, der technische Lösungen für seine Kunden entwickelt und diese bereitstellt. Dieses eher klassische »1:n«-Modell verschiebt sich mit einer Vereinfachung von Standards und mit der Übernahme von Technologien und Designprinzipien aus dem Web 2.0-Bereich hin zu loseren, flexibleren Wertschöpfungsketten, die im letzten Kapitel dieses Artikels aufgezeigt werden.

2.5 Erlösmodelle

Erlösmodelle im Bereich Web 2.0 unterscheiden sich vor allem aus zwei Gründen von traditionellen Ansätzen in der Softwareindustrie: Anwendungen werden nicht als statisches Artefakt über klassische Kanäle im Einzelhandel verkauft, sondern vielmehr als Dienst im Internet angeboten. [O'Reilly 2007] beschreibt dieses Phänomen als »Ende des Software-Releasezyklus« und proklamiert das Konzept des »Software as a Service« (SaaS). Die Suchmaschine »Google« beispielsweise steht Nutzern ununterbrochen als Dienst zur Verfügung, während im Hintergrund Algorithmen und Datenbankstrukturen angepasst werden können. Der zweite Grund für neue Erlösmodelle im Web 2.0-Kontext ist die Abhängigkeit der verschiedenen Dienste von der Nutzeranzahl. Oft ist es nicht nur die angebotene Funktionalität, sondern vor allem die von Nutzern selbst generierten und gesammelten Daten, die ihre Attraktivität ausmachen. Viele Anbieter schrecken daher vor der Einführung von Nutzungsgebühren zurück, weil sie eine Reduzierung der Nutzeranzahl und damit einen Verlust der Attraktivität des Dienstes fürchten. Die Integration von Werbung Dritter und die Einführung von Nutzergebühren für Premiumdienste stellen profitable Ausnahmen dar. Im Falle von SOA-Anwen-

dungen werden technische Lösungen noch meist als Artefakte mit bestimmten Lizenzgebühren verkauft und neue Versionen in bestimmten Zyklen gegen Zahlung weiterer Gebühren angeboten.

2.6 Warenfluss

Im Bereich Web 2.0 ist die Produktherstellung von starker Nutzerpartizipation und dem oben erwähnten ständigen Beta-Zustand geprägt. Individuen werden in die Produktgestaltung explizit mit einbezogen und als essenzieller Teil des eigentlichen Mehrwerts betrachtet. Web 2.0-Anwendungen werden außerdem selten über klassische Kanäle an Kunden vertrieben: Oftmals werden neuartige Methoden wie virales Marketing [O'Reilly 2007], also sich selbst von Nutzer zu Nutzer verbreitende Werbung, eingesetzt. Im SOA-Bereich werden Produkte meist wie zuvor beschrieben als Artefakte angesehen, von wenigen Experten exklusiv hergestellt und über traditionelle Kanäle vertrieben.

2.7 Einflüsse des sozialen Umfelds

Gesetzliche Bestimmungen werden für Web 2.0 zu einem Faktor von wachsender Bedeutung, weil sie die bisherige Dynamik der Anwendungen einschränken könnten. Zwar können Inhalte und Funktionalitäten aus technischer Perspektive auf fast beliebige Weise genutzt, komponiert und wieder angeboten werden, jedoch existiert eine wachsende Zahl von Fällen, in denen Betreiber von Blogs oder Plattformen verpflichtet werden, die von ihnen veröffentlichten Ressourcen auf mögliche Gesetzesverletzungen hin zu überprüfen [Högg et al. 2006]. Das Unternehmensumfeld, in dem SOA-Anwendungen meist eingesetzt werden, fordert die Einhaltung von inflexiblen Gesetzeswerken wie dem *Sarbanes-Oxley Act* und unterliegt komplexen Governance-Prozessen [Weill & Ross 2004]. Die Realisierung der eigentlich angestrebten losen Kopplung und der schnellen Adaptierbarkeit wird dadurch erheblich eingeschränkt.

Business-Modell-Komponente	Web 2.0	SOA
Gestaltung des Produkts	Betonung der <i>Wiederverwendbarkeit</i> und Komposition existierender Module, <i>Kopplung</i> von entfernten und heterogenen Ressourcen, hohe <i>Agilität</i> als Designziel, » <i>User Empowerment</i> «	
	Anwendungen fokussieren sich meist auf die Präsentationsschicht und den Nutzer (»rich user experience«)	»Soziale Aspekte« fehlen Hoher Grad an Ex-ante-Bestimmung Strikte Governance-Prozesse
Technologien	Kopplung entfernter Ressourcen über <i>einheitliche Schnittstellen</i> , <i>Information Hiding</i> , <i>Reduzierung</i> von <i>Programmieraufwand</i>	
	Funktionale Restriktion auf technischer Ebene Syndizierung Geringe Komplexität	Viele Design-Freiheitsgrade, aber Problem der semantischen Interoperabilität Koordination/Choreographie Hohe Komplexität
Kunden	Individuen/kleine Firmen (»Long Tail«): Selbstverwirklichung, soziale Anerkennung und das Knüpfen von Beziehungen als Nutzen	Mittlere oder große Firmen, die interne Applikationslandschaften harmonisieren und Agilität verbessern möchten
Wertschöpfungsketten	Aufbrechen traditioneller Wertschöpfungsketten: lose Netzwerke von Anbietern/Konsumenten	»Experten« liefern und pflegen Software-»Artefakte« und werden oft durch Dritte dabei unterstützt
Erlösmodelle	Wenige erfolgreiche Kommerzialisierungsstrategien (z. B. Werbung/Mitgliedsgebühren)	Traditionelle Modelle: Verkauf von Lizenzen für Software-»Artefakte«
Warenfluss	Ständiger »Beta-Zustand« und Ende des Software-Releasezyklus	Zyklische Herstellung neuer Produktversionen
	Zunehmende Reduzierung der Agilität durch externe Einflüsse wie gesetzliche Bestimmungen	
Einflüsse des sozialen Umfelds	Fördert die Vernetzung von Menschen	Dient der Verbesserung von Effizienz und Agilität von Anwendungen

Tab. 1: Zusammenfassung der vergleichenden Analyse

3 Konvergenz der beiden Konzepte: Das Internet der Dienste

Wie aus Tabelle 1 ersichtlich, sind typische Web 2.0-Anwendungen dafür konzipiert, die Bedürfnisse von Menschen zu bedienen. Sie müssen einfach zu nutzen sein und die schnelle Komposition von Inhalten oder Applikationen ohne Berücksichtigung vieler formaler Vorschriften erlauben. SOA-Anwendungen hingegen unterliegen aufgrund ihrer Verwendung im Unternehmenskontext strikten Regelwerken und in-flexiblen Standards, um die fehlerfreie Funktion

von Geschäftsprozessen sicherzustellen. Trotz dieser zunächst unterschiedlichen Ausrichtung entwickelt sich ein Trend der Kombination von Technologien, Designprinzipien und Konzepten aus beiden Welten.

Das sogenannte *Internet der Dienste* ist eine Bezeichnung für eine neue Generation des Internets, das als globale Plattform für das Auffinden, Kombinieren, Benutzen und Veröffentlichlichen von interoperablen, elektronischen Diensten dienen wird. Es wird die Interaktion sowohl von Individuen, mittleren und großen Unternehmen verändern und sich dabei auf Designprin-

zipien und Technologien aus den Bereichen Web 2.0 und SOA stützen: Das Bedürfnis von Individuen sowie von Firmen zu kollaborieren, wird dabei zunächst durch *serviceorientierte Architekturen* unterstützt. Im Unternehmenskontext können organisationsübergreifende Prozesse durch die Orchestrierung und lose Kopplung von Services in einer schnell modifizierbaren Weise realisiert werden. Darüber hinaus werden Marktplätze zum Handeln von Diensten entstehen, die Nutzern deutlich mehr Funktionalität anbieten, als im konventionellen Web-Service-Stack vorgesehen ist. Neben der Serviceorientierung müssen dabei zentrale Elemente aus dem *Web 2.0*-Bereich berücksichtigt werden, um die Auffindbarkeit und Benutzung von Services durch menschliche Nutzer zu verbessern. Sogenannte »Folksonomies« entstehen, wenn Nutzer mithilfe von Tags (Schlagwörtern) internetbasierte Ressourcen beschreiben und diese Beschreibungen auf zentralen Plattformen gesammelt werden. Bei einer ausreichend großen Anzahl dieser Beschreibungen können diese eine repräsentative Meinung widerspiegeln. Suchen Nutzer dann nach bestimmten Inhalten oder Funktionalitäten und bedienen sich dabei der hinterlegten Hinweise, dann können gesuchte Ressourcen schnell identifiziert werden. Sogenannte »Gadgets« werden dazu dienen, Services ein Gesicht zu geben, Nutzern also die zugrundeliegende Funktionalität auf intuitive Weise zu veranschaulichen und dadurch die Interaktion zwischen menschlichen Individuen und elektronischen Services zu revolutionieren. Über neu entstehende Plattformen wird es Individuen möglich sein, solche Gadgets zu finden, zu bewerten (»taggen«), dynamisch zu komponieren und auf diese Weise »Mashups« mit neuer, individuell zusammengestellter Funktionalität zu kreieren. Die explizite Integration des wenig technikaffinen Nutzers in die Gestaltung und Bewertung von webbasierten Applikationen, die dem Web 2.0-Gedanken Rechnung trägt, wird so dazu beitragen, eine globale serviceorientierte Architektur zu realisieren.

Die Idee des *Mashup-Unternehmens* stellt ein konkretes Beispiel für eine kombinierte Anwendung von Konzepten aus den Bereichen Web 2.0 und SOA dar [Mulholland et al. 2006]. Anders als im Falle klassischer SOA-Implementierungen, die auf komplexen Protokollen aufbauen und von wenig technisch orientierten Nutzern nicht bedient oder gar modifiziert werden können, zielen die Konzepte des Mashup-Unternehmens darauf ab, gerade diese Nutzer in den Prozess des Designs von elektronischen Geschäftsbeziehungen zu involvieren. Dies steigert unternehmerische Agilität und ermöglicht es, modulare Dienstleistungen global zu suchen, zusammzusetzen und diese in das eigene Geschäftsmodell mit minimalem Aufwand zu integrieren. Die Firma Kapow Technologies (www.kapotech.com) hat kürzlich eine sogenannte Web-Integration-Plattform vorgestellt, die eine erste, technische Lösung für die Erstellung des oben vorgestellten Mashup-Unternehmens bietet. Sie ermöglicht es Nutzern ohne signifikanten Programmieraufwand, internetbasierte Anwendungen zusammenzustellen und zu koordinieren, um Geschäftsmodelle schnell umsetzen zu können.

4 Schlussbemerkungen

Im Rahmen dieses Artikels wurde ein systematischer Vergleich der beiden Konzepte Web 2.0 und SOA auf der Grundlage von sieben sowohl technischen als auch betriebswirtschaftlich geprägten Kategorien durchgeführt. Dabei wurden deutliche Unterschiede identifiziert, die sich aus den unterschiedlichen Umfeldern der Verwendung ergeben. Auf der anderen Seite folgen die Anwendungsbeispiele beider Konzepte oft gleichen Prinzipien (Wiederverwendung existierender Ressourcen, Komposition, lose Kopplung, Komplexitätsreduktion, Agilität, »User Empowerment«). Der inzwischen sehr komplexe Web-Service-Stack ist für die Entwicklung einer globalen, serviceorientierten Architektur, also eines *Internets der Dienste*,

nicht adäquat. Mit WSDL beschriebene Web-Services sind für wenig technikaffine Nutzer schwer auffindbar und benutzbar. Im Internet der Dienste konvergieren Technologien und Designprinzipien aus beiden Welten: Services erhalten hier erstmals ein »Gesicht« und sind intuitiv bedien- und kombinierbar, was in konventionellen SOA-Anwendungen nicht gegeben ist. Herkömmlichen Web 2.0-Anwendungen fehlen dagegen oftmals Struktur und Kontrollmechanismen, um im Unternehmenskontext Fuß zu fassen und den dort existierenden gesetzlichen Regelungen gerecht zu werden. Durch die zunehmende Konvergenz beider Welten werden sich zukünftig ganzheitliche Anwendungen schaffen lassen, die sowohl im Unternehmenskontext als auch außerhalb von Unternehmensgrenzen großen ökonomischen Mehrwert zu leisten im Stande sind.

5 Literatur

- [Alonso et al. 2004] *Alonso, G.; Casati, F.; Kuno, H.; Machiraju, V.*: Web Services Concepts, Architectures and Applications. Springer-Verlag, Berlin, 2004.
- [Anderson 2006] *Anderson, C.*: The Long Tail: Why the Future of Business is Selling Less of More. Hyperion Books, 2006.
- [Fielding 2000] *Fielding, R. T.*: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine, 2000.
- [Högg et al. 2006] *Högg, R.; Meckel, M.; Stanoevska-Slabeva, K.; Martignoni, R.*: Overview of business models for Web 2.0 communities. In: Proceedings of GeNeMe 2006, Dresden, 2006, S. 23-37.
- [McAfee 2005] *McAfee, A.*: Will Web Services Really Transform Collaboration. MIT Sloan Management Review, Vol. 46, No. 2, 2005.
- [McAfee 2007] *McAfee, A.*: Enterprise 2.0 vs. SOA. In: Harvard Business School Faculty Blog, <http://blog.hbs.edu/faculty/amcafee/index.php>; Zugriff am 20.01.2007.
- [Mulholland et al. 2006] *Mulholland, A.; Thomas, C. S.; Kurchina, P.*: Mashup Corporations: The End of Business As Usual. Evolved Technologist Press, 2006.
- [O'Reilly 2007] www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-2.0.html; Zugriff am 20.01.2007.
- [Timmers 1998] *Timmers, P.*: Business Models for Electronic Markets. International Journal on Electronic Markets and Business Media, Vol. 8, No. 2, 1998.
- [UN/CEFACT 2007] www.unece.org/cefact/; Zugriff am 20.01.2007.
- [Weill & Ross 2004] *Weill, P.; Ross, J. W.*: How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, Boston, 2004.

Dipl.-Ing. Christoph Schroth
SAP Research CEC
Universität St. Gallen
Institut für Medien und
Kommunikationsmanagement
Blumenbergplatz 9
CH-9000 St. Gallen
christoph.schroth@unisg.ch
www.mcm.unisg.ch

Dipl.-Ing. Inf. (BA) Dipl.-Des. Lars Kirchhoff
Universität St. Gallen
Institut für Medien und
Kommunikationsmanagement
Blumenbergplatz 9
CH-9000 St. Gallen
lars.kirchhoff@unisg.ch
www.mcm.unisg.ch