

Evolutionäre betriebliche Informationssysteme - Perspektiven und Herausforderungen einer neuen Generation von Informationssystemen¹

Gustaf Neumann, Stefan Sobernig, Michael Aram
Institut für Wirtschaftsinformatik und Neue Medien,
Wirtschaftsuniversität Wien,
Welthandelsplatz 1, 1020, Wien, Österreich

Der vorliegende Beitrag konzentriert sich auf bestehende und zukünftige Herausforderungen „evolutionärer betrieblicher Informationssysteme“, einer Gattung von Systemen, die einen evolutionären Softwareentwicklungsprozess erfordern und die „sekundäres Design“ auf mehreren konzeptionellen Ebenen unterstützen. Die Arbeit ordnet bestehende Forschungsbeiträge und zukünftige Herausforderungen innerhalb einer idealisierten, vorläufigen Systemarchitektur ein. Die Arbeit betont eine pluralistische Sichtweise auf den Forschungsgegenstand und leitet daraus die Notwendigkeit einer methodologischen Flexibilität im Sinne interdisziplinärer Konfigurationen von Forschungsmethoden ab.

Schlagerworte: Evolutionäre betriebliche Informationssysteme, Sekundäre Gestaltung, Pluralistische Wirtschaftsinformatik

1 Einleitung

Dieser Beitrag beschreibt das Forschungsfeld „Evolutionäre betriebliche Informationssysteme“, das sich innerhalb eines Verständnisses von Wirtschaftsinformatik als sozialwissenschaftliche Disziplin entwickelt und das Ziel verfolgt, die betriebliche Leistungsfähigkeit zu verbessern (Hansen und Neumann 2009). Diese soziotechnische Sichtweise wird auch als „ensemble view of technology“ (Orlikowski und Iacono 2001) bezeichnet.

Informationstechnologie kann Transaktions- und Koordinationskosten drastisch reduzieren und führt häufig zu signifikanten Änderungen in der Art wie Unternehmen agieren. Dementsprechend bezeichnen Malone et al. (1999) Informationssysteme als „tools for inventing organizations“. Da technologische Artefakte (March und Smith 1995) die Gestaltungsräume eines betrieblichen Informationssystems formen, ist es wichtig, sie nicht nur isoliert zu analysieren, sondern diese und deren zu Grunde liegenden Designentscheidungen im Kontext eines im Einsatz befindlichen Informationssystems zu studieren (siehe Abb. 1).

¹ Die finale Version dieses Manuskripts ist in deutscher und englischer Sprache über <http://www.springerlink.com> und <http://www.bise-journal.org> erschienen: Neumann G, Sobernig S, Aram M (2014) Evolutionary Business Information Systems. Perspectives and Challenges of an Emerging Class of Information Systems. Bus Inf Syst Eng. doi: 10.1007/s12599-013-0305-1.

Siehe <http://link.springer.com/article/10.1007/s11576-013-0397-8>

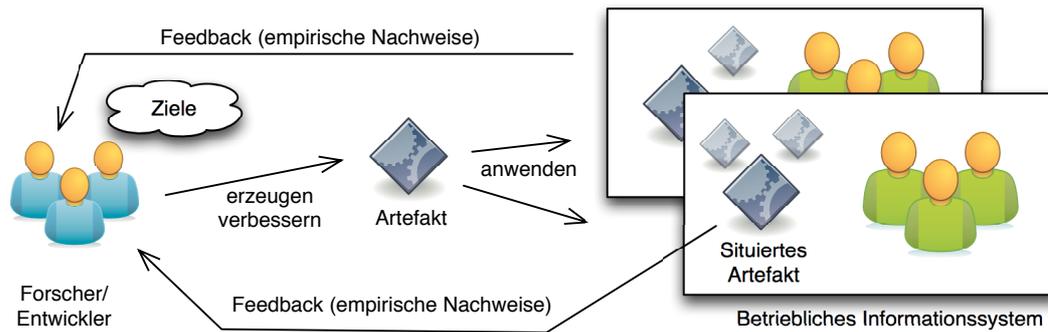


Abb. 1 Primärer Entwicklungszyklus mit Feedback-Schleife

2 Evolutionäre betriebliche Informationssysteme

Informationssysteme größerer Organisationen entwickeln sich ständig weiter, um sich der im stetigen Wandel befindlichen Unternehmenssituation anzupassen. Dementsprechend sehen sich diese Systeme mit einem permanenten Nachholbedarf in ihrem Entwicklungsprozess konfrontiert. Schon in den 1990er Jahren stellten Allen und Boynton (1991, S. 435) fest: „IS efforts generally automate the status quo, freezing the organization into patterns of behavior and operations that resolutely resist change“. Betriebliche Informationssysteme sind demnach (im Sinne der Erfüllung sämtlicher Geschäftsanforderungen) nie fertiggestellt (Bjerknes et al. 1991).

Es wurden zahlreiche Ansätze in einer Reihe von Forschungsfeldern entwickelt, um diesen charakteristischen Entwicklungsrückstand zu mindern, von partizipativem Design (Muller und Kuhn 1993) über agile Softwareentwicklungsmethoden (Abrahamsson et al. 2010) zu modellgetriebener Softwareentwicklung (Stahl et al. 2006) und Software-Produktlinien (Clements und Northrop 2007). Obwohl diese Ansätze zu beachtlichen Verbesserungen in den jeweiligen Anwendungsbereichen geführt haben, fehlt der Wirtschaftsinformatik eine ganzheitliche Sicht – in der betrieblichen Praxis besteht der angesprochene Entwicklungsrückstand nach wie vor.

Die Herausforderung evolutionärer betrieblicher Informationssysteme ist es, eine soziotechnische Systeminfrastruktur zu schaffen, die eine fortlaufende, unterbrechungsfreie Weiterentwicklung und Anpassung von Informationssystemen an sich verändernde Geschäftsanforderungen unterstützt. Hierzu müssen Stakeholder mit unterschiedlicher Expertise in die Lage versetzt werden, unter Umständen unvorhergesehene Änderungen direkt und inkrementell in das Informationssystem einzubringen. Dies sollte ohne Dienstunterbrechung möglich sein. Es sei angemerkt, dass sich der Terminus „evolutionäre betriebliche Informationssysteme“ auf das gesamte Informationssystem bezieht, aber nicht notwendigerweise auf einzelne softwarebasierte Anwendungen oder softwaregestützte Prozesse. Evolutionäre betriebliche Informationssysteme können Selbstkoordination durch nicht-hierarchische Kommunikation unterstützen (Kieser und Kubicek 1992). Selbstkoordination kann existierende Organisationsstrukturen permanent infrage stellen, um eine Organisation an veränderliche Anforderungen anzupassen.

Folglich sollte eine kontinuierliche Organisationsentwicklung durch das Informationssystem unterstützt werden.

Ein Charakteristikum evolutionärer betrieblicher Informationssysteme ist „sekundäres Design“ (Germonprez et al. 2011), durch das die Benutzer eines anpassbaren Informationssystems die zentralen Akteure dessen kontinuierlicher Gestaltung werden. Dabei verändern die Benutzer das System im Rahmen ihrer Systemnutzung – häufig ohne Kenntnis der ursprünglichen, primären Designentscheidungen. Das Konzept vom „sekundären Design“ berücksichtigt, dass sich Nutzerverhalten und Geschäftskontext mit der Zeit verändern und dass Informationssysteme durch ihre Benutzer besetzt werden, indem diese das Informationssystem für die Erledigung der ihnen zugeteilten Arbeitsaufgaben modifizieren. Der primäre Designer gibt die zentrale Kontrolle über die Gestaltung auf und erlaubt benutzergetriebene Innovation (von Hippel 2009). Dementsprechend sollten betriebliche Informationssysteme als höchst modifizierbare Softwaretechnologie (Germonprez et al. 2007) ausgestaltet sein, um aufgabenbezogene Handlungen, Handlungsreflexion und dafür notwendige Systemmodifikationen eines Benutzers zu unterstützen. Ein solches System darf jedoch nicht allen Benutzern so weitreichende Freiheitsgrade wie zum Beispiel ein konventionelles Wiki-System einräumen, in welchem jedem Benutzer beliebige Änderungen gestattet sind. Vielmehr sollte es, wie etwa ein Enterprise-Wiki-System, Lenkungsinstrumente auf Basis funktionaler Verantwortlichkeitsbereiche und organisationaler Rollen vorsehen.

Wünschenswerte Funktionalitäten von evolutionären betrieblichen Informationssystemen umfassen:

- Mechanismen für das kontinuierliche Einbringen inkrementeller (Ad-hoc-)Änderungen
- Funktionen für das Ermöglichen des sekundären Designs von Inhalten und Systemverhalten durch eine Vielzahl von Benutzern
- Unterstützung von Systemintrospektion zur Schaffung von Feedback-Kanälen über den aktuellen Systemstatus, das Systemverhalten und vorangegangene Benutzeraktionen
- Mechanismen zur Behandlung einer Vielzahl von Versionen und Varianten der Instanzen und Schemata
- Mechanismen zum Abgleich von System-/Organisationsanforderungen (Governance, Stabilität) und individuellen Bedarfen (Flexibilität, Anpassbarkeit)

Der systematische Entwicklungszyklus primären Designs, wie in Abbildung 1 dargestellt, ist zweifelsohne weiterhin notwendig, er ist jedoch nicht mehr der einzige Mechanismus zur Erweiterung und zur Anpassung des betrieblichen Informationssystems.

Viele dieser wünschenswerten Eigenschaften sind in heutigen Systemen bereits verfügbar. So unterstützen traditionelle Wiki-Systeme bereits sekundäres Design, allerdings vorrangig auf der Inhaltsebene. Forschung zu Ad-hoc-Workflows (Georgakopoulos et al., 1995) fokussiert sich auf die Unterstützung von außerplanmäßigen Aktivitäten, die eine zeitnahe und rasche systemgestützte Ausführung erfordern. Derartige Ad-hoc-Änderungen betreffen üblicher-

weise Workflow-Instanzen anstatt Workflow-Schemata. Enterprise-Mashup (EM)-Systeme sind auf die Programmierung durch Endbenutzer ausgelegt und erlauben den Benutzern personalisierte, situative Anwendungen zu erstellen, welche ihre unmittelbaren Geschäftsanforderungen adressieren (Pahlke et al. 2010).

Ein Ziel von evolutionären betrieblichen Informationssystemen ist es, nicht nur Ad-hoc-Änderungen auf Instanzebene zu unterstützen, sondern Domänenexperten das Ändern und das Wiederverwenden von Schemadefinitionen direkt zu ermöglichen, etwa um eine neue, abgeleitete Klasse von Geschäftsprozessen zu unterstützen. Dafür benötigen sie ausreichend operationales Wissen, um die systemweiten Konsequenzen ihrer gestalterischen Handlungen abzusehen. Je mehr Benutzer befähigt werden, das System über den engen Bereich der persönlichen Nutzung hinaus zu adaptieren, desto wichtiger ist es, die Transparenz des Systemverhaltens zu erhöhen (Breu et al. 2011). Demnach müssen Artefakte in hohem Maße Introspektion erlauben und Gestaltungsaktivitäten müssen für die Nutzer nachvollziehbar gemacht werden, wobei Stakeholder mit unterschiedlichem Fachwissen unterstützt werden müssen. In diesem Sinne müssen evolutionäre betriebliche Informationssysteme laufend empirische Information über das System- und Geschäftsverhalten liefern, um für die Stakeholder (die Systemgestalter) eine Basis für die Analyse und Fortentwicklung bereit zu stellen.

Wenn viele Domänenexperten laufend ein System kollektiv entwickeln, steigt zwangsläufig die Vielfältigkeit des Systems. Ein evolutionäres betriebliches Informationssystem muss mit einer Vielfalt von Versionen bzw. Varianten (von bspw. Arbeitsschritten, Workflows, und Geschäftsprozessen) in einer skalierbaren Form umgehen können (vgl. dazu das Fallbeispiel in Sektion 3). Schließlich, wenn Ad-hoc-Änderungen mehrere Anwendungen – und nicht lediglich eine einzelne Instanz (Ad-hoc-Workflows) oder einzelne Benutzer (EM-Systeme) – beeinflussen, dann ist es entscheidend, geeignete Lenkungsmechanismen bereitzustellen, welche die Änderbarkeit bestimmter Systemeigenschaften einschränken sowie Verlässlichkeit und Vorhersehbarkeit sicherstellen.

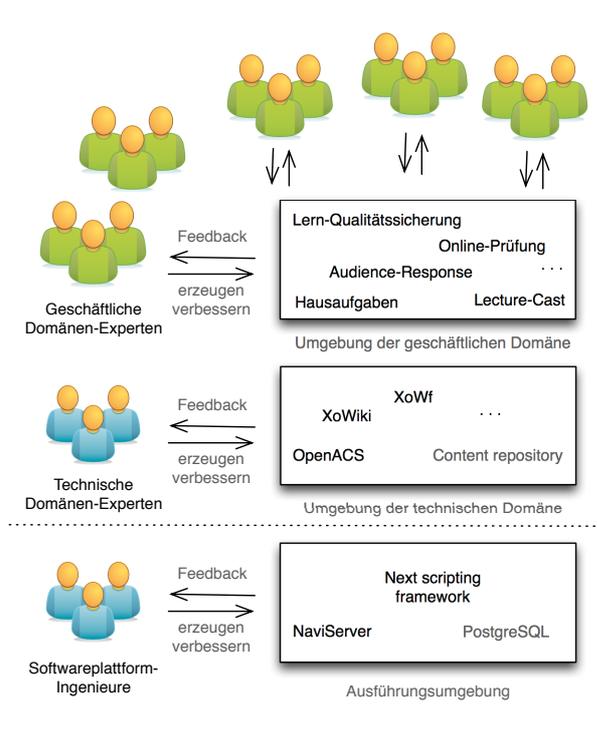
Tab. 1 Systemarchetypen und ihre evolutionären Eigenschaften

| | Wiki | EM-Systeme | Ad-hoc-Workflow-Systeme | Evolutionäre betriebliche IS |
|--------------------------|------|------------|-------------------------|------------------------------|
| Ad-hoc-Änderungen | + | + | + | + |
| Sekundäres Design | + | + | - | + |
| Inhaltsentwicklung | + | + | - | + |
| Instanzenentwicklung | + | + | + | + |
| Schemaentwicklung | - | - | - | + |
| Variabilitätsmanagement | + | - | + | + |
| Kontrollflussanpassungen | - | - | + | + |
| Feedback-Kanal | + | - | - | + |
| Governance-Unterstützung | - | - | - | + |

Der folgende Abschnitt beschreibt ein großes Informationssystem im Bereich E-Learning, welches in ein evolutionäres betriebliches Informationssystem weiterentwickelt werden soll.

3 Ein Fallbeispiel

Ein erfolgreicher Forschungsansatz der Wirtschaftsinformatik ist es, Artefakte in situ zu entwickeln, um einen Rückkanal und Messgrößen für die Evaluation der Designentscheidungen zu erhalten. Wir ziehen im Folgenden Learn@WU (Alberer et al. 2003) als ein Fallbeispiel heran, welches bereits viele der dargelegten Eigenschaften evolutionärer betrieblicher Informationssysteme aufweist. Es ist eines der größten universitären E-Learning-Systeme in Hinblick auf dessen Nutzung mit über 160.000 Lernressourcen, bis zu 4 Mio. Seitenaufrufen pro Tag und bis zu 2.500 gleichzeitigen Benutzern.



Open-Source-Softwareartefakte in Learn@WU:

- Workflow-System, das Ad-hoc-Änderungen von Schemata und Instanzen erlaubt (XoWf)
- Enterprise-Wiki-System (XoWiki)
- Communitybasierte Plattform, die dezentralisierte Entwicklung durch nicht technische Nutzer ermöglicht (OpenACS)
- Dynamische Skriptsprache, Sprachunterstützung für Softwareentwurfsmuster und Softwareevolution (Next Scripting Framework, Extended Object Tcl)
- Skalierbarer Webserver mit eingebetteter Skriptsprache, der inkrementelle Softwareevolution und umfangreiche Systemüberwachung unterstützt (NaviServer)

Abb. 2 Beispiele von Artefakten und Stakeholdern in der mehrschichtigen Learn@WU-Architektur

Zu den Stakeholdern von Learn@WU gehören verschiedene Gruppen von Domänenexperten (z.B. Hochschullehrer, E-Learning-Assistenten, Programmdirektoren oder die Qualitätssicherung der Lehre), welche die Interaktionen mit und zwischen den Lernenden gestalten, sowie Lerninhalte und Lernanwendungen entwickeln. Ebenso gehören zu den Stakeholdern systemtechnische Domänenexperten. Die Eigenschaften der entwickelten Systemkomponenten erlauben die laufende Verbesserung anwendungsspezifischer Aspekte des Systems ohne Dienstunterbrechungen. Das System unterstützt eine dezentrale Entwicklung, bei der die Stakeholder ihre Gestaltungsräume auf Basis von vorgefertigten Komponenten konfigurieren und weiterentwickeln können. Die Menge der bereitgestellten Komponenten wird laufend erweitert und basiert auf einer flexiblen Content-Management- und Workflow-Infrastruktur, die mit Skriptsprachen an die Aufgabenstellungen angepasst werden. Ein aktuelles Beispiel einer der-

artigen Anpassung ist eine Audience-Response-Komponente (Andergassen et al. 2012) zur Erfassung von Feedback von Studierenden innerhalb von Lehreinheiten auf Basis eines Wiki-basierten Workflow-Systems (Neumann und Erol 2008).

Je mehr Stakeholder tatsächlich das laufende System verändern, desto breiter ist die im System herrschende Variantenvielfalt, sowohl auf der Inhalts- als auch auf der Verhaltensebene. Auf der Inhaltsebene könnte das sekundäre Design in den Wiki-Systemen der Stakeholder ähnlich wie bei Germonprez et al. (2011) untersucht werden. Darüber hinaus ist es jedoch auch interessant, die beobachtbare Vielfalt der Workflow-Definitionen und -Instanzen als Indikator für das sekundäre Design auf Verhaltensebene zu quantifizieren. Derzeit nutzt das System 636 unterschiedliche Workflow-Definitionen, welche von 59 verschiedenen Benutzern ko-entwickelt worden sind. Für diese Definitionen existieren derzeit 1.417 Revisionen. Im System existieren derzeit über 500.000 Workflow-Instanzen mit über 2,5 Mio. Rückverfolgungspunkten. Mehr als 20.000 Benutzer haben diese Workflow-Instanzen verwendet. Diese Zahlen unterstreichen die Notwendigkeit für ein skalierbares Variabilitätsmanagement, um bspw. die Nachvollziehbarkeit, die Folgenabschätzung von Veränderungen und die Wiederverwendung von kodifiziertem Wissen im großem Umfang zu ermöglichen.

Unsere Erfahrung mit Learn@WU zeigt, dass der Einsatz von Wiki-basierten Workflow-Systemen zu höherer Produktivität der Entwickler und höherer Flexibilität der Komponenten führt. Im Fall von Learn@WU stehen sechs Mitarbeitern für Betrieb und Entwicklung etwa zehn mal so viele Personen gegenüber, die zu den Workflow-Definitionen beigetragen haben. Wir sind zuversichtlich, dass dieses Verhältnis noch weiter erhöht werden kann. Zusätzlich erlaubt uns die domänenspezifische Transaktionsüberwachung schon jetzt, die systemgestützten Lernaktivitäten der Studierenden besser zu analysieren (Mödritscher et al. 2013).

4 Industrielle Anwendungen

Das Feld der evolutionären betrieblichen Informationssysteme gründet auf Ideen der Endbenutzerpartizipation und erweitert diese um Konzepte aus den Bereichen evolvierender Softwaresysteme und sekundären Designs. Die Industrie hat bereits begonnen, diese Konzepte aufzunehmen. Enterprise-Wiki-Systeme wie etwa Confluence werden in tausenden Unternehmen dazu verwendet, die Zusammenarbeit und den Wissensaustausch zu verbessern. Endbenutzerpartizipation, die über die Inhaltsebene hinaus geht, ist in heutigen Informationssystemen weniger weit verbreitet, hat aber für einzelne Teilaufgaben ihren Weg in die Produktportfolios großer Software-Anbieter gefunden. Beispielsweise hat SAP Research den Forschungsprototyp *Rooftop Marketplace* (Hoyer et al. 2009) entwickelt, der ein Enterprise-Mashup-System realisiert. Oracle bietet EM-Funktionalität als Teil der *WebCenter Suite* an. IBM und Software AG haben ihre EM-Ansätze in die Softwareprodukte *IBM Mashup Center* und *ARIS MashZone* eingebracht. Das Konzept domänenspezifischer Softwaresprachen (DSL; Fowler 2010) wurde eingeführt, um die Kommunikation zwischen nicht-technischen Domänenexperten und Entwicklern zu ermöglichen. Deren erfolgreicher Einsatz ist jedoch häufig durch das Fehlen verlässlichen Domänenwissens als Grundlage der DSL-Entwicklung erschwert (Mernik et al. 2005).

Die Softwarekomponenten von Learn@WU haben bereits das Interesse von Industrie und öffentlichen Einrichtungen geweckt. Daimler AG verwendet beispielsweise diese Komponenten zur Unterstützung des Wissensmanagements im Supply-Chain-Management (Unternehmen und Zulieferer). LMS.at nutzt diese um mehr als 2.600 österreichische Schulen der Sekundarstufe zu unterstützen.

5 Skizze einer Forschungsagenda

Das übergeordnete Forschungsziel ist es, Methoden und Instrumente zu entwickeln, die die direkte und unmittelbare Anpassbarkeit betrieblicher Informationssysteme durch die Stakeholder verbessern, die aber auch Organisationen Lenkungsformen für die Einhaltung von Grundsätzen ermöglichen und die Systemstabilität gewährleisten. Die grundlegenden Stoßrichtungen in diesem Forschungsfeld sind:

1. Wie kann man systematisch Verbesserungspotenzial in den wachsenden Gestaltungsräumen identifizieren?
2. Wie kann ein laufendes betriebliches Informationssystem inkrementell weiterentwickelt werden?
3. Wie kann man den Partizipationsgrad nicht technischer Stakeholder im (sekundären) Gestaltungsprozess des betrieblichen Informationssystems erhöhen?

Für jede dieser Stoßrichtungen müssen unterschiedliche Forschungsmethoden angewandt werden. Für (1) stellen die von den In-situ-Artefakten gewonnenen empirischen Werte die primäre Analysebasis dar. Die systematische Analyse von Transaktionsdaten ist im Bereich Business Analytics bereits etabliert, jedoch hat sich der Fokus in jüngster Zeit auf die Analyse des Systemverhaltens verschoben. Hierfür stellt die Koordinationsforschung (Coordination-Science) Techniken zur Verfügung (Malone und Crowston 1994), besonders wenn Transaktionsdaten mit externen (Sensor-)Daten kombiniert werden. Beispiele sind das Process-Mining (van der Aalst 2011), datengetriebene Entscheidungsfindung (Brynjolfsson et al. 2011) und Learning-Analytics (Siemens und Long 2011) im Bereich E-Learning.

Stoßrichtung (2) adressiert die Anforderung, die Flexibilität der Informationssysteme zu erhöhen und zielt auf leichtgewichtige Entwicklungszyklen auf technischer und organisatorischer Ebene ab: Ein unternehmensweites Softwaresystem, das laufend von einer Vielzahl von Benutzern für die tägliche Arbeit genutzt wird, erlaubt nicht, dass die Software bei hunderten täglichen Aktualisierungen nach jeder Änderung rekompiliert und neugestartet werden muss. Das Ziel ist es, auf evolutionäre Systeme hinzuarbeiten, die Selbstorganisation unterstützen und ihr Verhalten im laufenden Betrieb anpassen können. Den State-of-the-Art in dieser Dimension voranzubringen verlangt nach Forschung in selbstorganisierenden sozialen Systemen (Wulf 1999), nach der Integration von Organisations- und Technologieentwicklung (Wulf und Rohde 1995) und nach Forschung zu Softwareflexibilität, insbesondere aber zu mehrschichtiger Softwareentwicklung (Ousterhout 1998), dynamischer Software-Evolution (Rank 2002), dynamischen Programmiersprachen (Callaù et al. 2011) und Software-Produktlinien (Clements und Northrop 2007).

Die Herausforderung von Stoßrichtung (3) ist es, die fortlaufende Partizipation von Domänenexperten zu fördern, um eine kollektive Entwicklung und Weiterentwicklung von informationsverarbeitenden Systemen direkt durch die Stakeholder zu ermöglichen, ohne dabei funktionsnotwendige wesentliche Eigenschaften zu verletzen. Dahingehend sollten die folgenden Aspekte eingehend beleuchtet werden:

- Geeignete Methoden zur Spezifikation von durch die Stakeholder modifizierbarem Verhalten, z.B. die systematische Konstruktion von domänenspezifischen Sprachen (Strembeck und Zdun 2009)
- Designprinzipien für die Gestaltung und die Kombination von wiederverwendbaren Softwarekomponenten, bspw. mittels Feature-orientierter Programmierung (Apel und Kästner 2009)
- Methoden zur systematischen Bereitstellung geeigneter Feedback-Kanäle für alle Stakeholder, welche auch Aspekte der IT-Sicherheit und des Schutzes der Privatsphäre adressieren; Methoden zu Data- und Process-Mining sowie Data- und Process-Monitoring durch Endbenutzer.
- Methoden zur skalierbaren Verwaltung von Prozess- bzw. Workflow-Varianten- und -Versionen, zur Schemaauswahl, Migration, Analyse und Überwachung eben dieser; Entwicklung von Entscheidungsunterstützungs- und Empfehlungssystemen basierend auf situativen Analysen.

Diese drei Stoßrichtungen sollten nicht isoliert, sondern gemeinsam, mit deren Interaktionen behandelt werden, was wiederum zu einer Verbesserung des State-of-the-Art in den einzelnen Dimensionen führen kann.

6 Interdisziplinäre, domänenspezifische Konfigurationen

Der soziotechnische Fokus der Informationssystemforschung verlangt eine ganzheitliche Sichtweise basierend auf verhaltens- und gestaltungsorientierten Wissenschaften (Hevner et al. 2011). Jüngste Bestrebungen versuchen etablierte gestaltungsorientierte Forschungsperspektiven (Peffer et al. 2007; Vaishnavi und Kuechler 2007) zu erweitern, um diesen Anforderungen zu entsprechen (Sein et al. 2011) und um Endbenutzer explizit als reflektierende und aktive Teilnehmer eines fortlaufenden (sekundären) Gestaltungsprozesses zu begreifen (Germonprez et al. 2011).

Die starke Betonung der Ausrichtung auf die jeweilige Anwendungsdomäne (beim genannten Beispiel: E-Learning) erfordert einen Fokus auf domänenspezifische Forschung, um passende Abstraktionen im Informationssystem bereitstellen zu können. Forschung im Bereich evolutionärer betrieblicher Informationssysteme verlangt somit im Allgemeinen ein pluralistisches Forschungsverständnis (Frank 2006). Unserer Erfahrung nach bedürfen sogar unterschiedliche Instanzierungen in derselben Domäne unterschiedliche Konfigurationen von Methoden aus verschiedenen Forschungsfeldern. Abbildung 3 skizziert relevante Forschungsfelder für die Erforschung des Informationssystems Learn@WU.

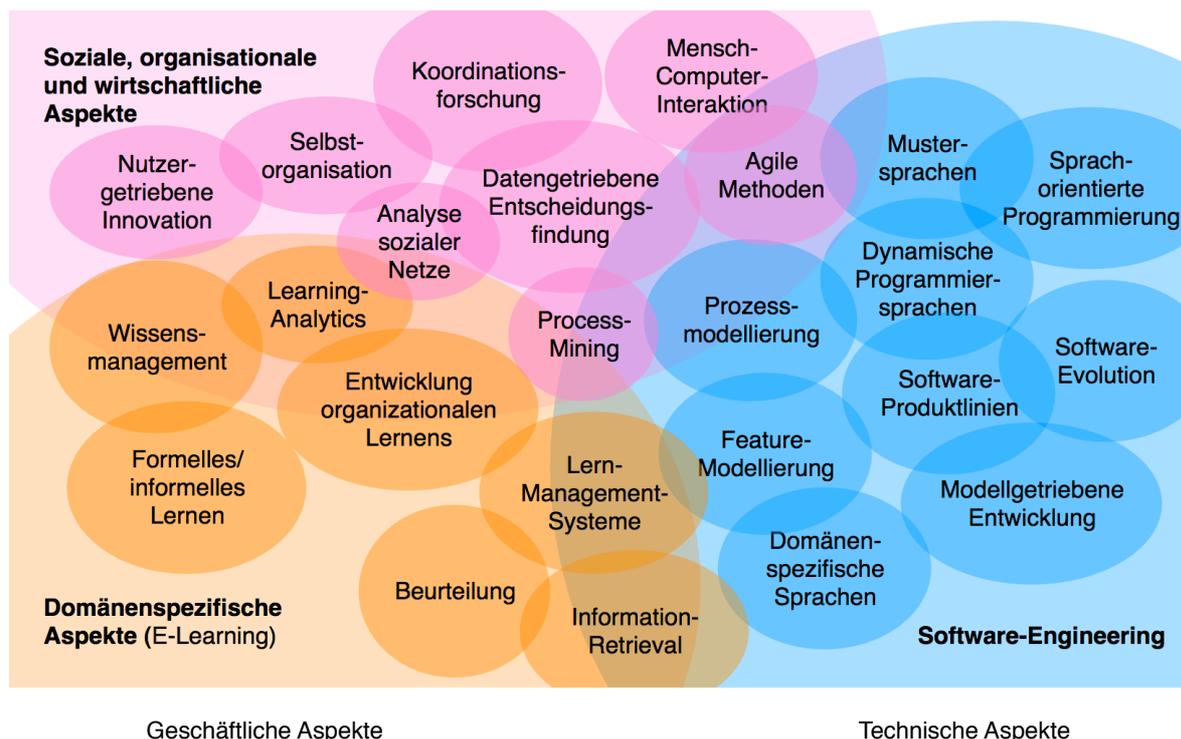


Abb. 3 Referenzdisziplinen für interdisziplinäre Forschung zu evolutionären betrieblichen Informationssystemen im Bereich E-Learning.

Diese pluralistische Sicht integriert verhaltensorientierte Forschungsaktivitäten zum Verständnis, zur Erklärung und zur Vorhersage von Phänomenen in existierenden, situierten Systemen mit konstruktionsorientierten Ansätzen zur Verbesserung des Status quo durch das Erstellen von innovativen Artefakten. Das Forschungsfeld verlangt nicht zwingend, dass jedem Forscher eine In-situ-Instanziierung zur Verfügung stehen muss. Gestaltungsorientierte Forschung kann dem Ansatz der Konsortialforschung folgen, um die Kooperation zwischen Forschern und Praktikern zu leiten (Österle und Otto 2010). Darüber hinaus kann das Feld von Forschungsergebnissen vieler Disziplinen profitieren, von der Informatik bis hin zu Sozialwissenschaften.

7 Zusammenfassung

In diesem Beitrag wurde das Konzept der evolutionären betrieblichen Informationssysteme herausgearbeitet, die eine laufende sekundäre Gestaltung der Informationssysteme durch Nutzer auf mehreren konzeptionellen Ebenen unterstützen. Entsprechende Systeme unterliegen ständiger Veränderung, welche von Stakeholdern mit heterogenem Domänenwissen und technischer Expertise in das System eingepflegt wird. Softwareartefakte, die in derartige Systeme eingebracht werden sollen, müssen eine kontinuierliche (sekundäre) Gestaltung und kontinuierliche empirische Evaluation unterstützen. Wir argumentieren, dass die Untersuchung evolutionärer betrieblicher Informationssysteme eine pluralistische Forschungssicht verlangt, da das Forschungsobjekt inhärent interdisziplinär ist. Die Wirtschaftsinformatik kann zu diesem aufstrebenden Feld innovative Artefakte beisteuern. Real im Einsatz befindliche Softwaresysteme fungieren hierbei als wesentliches Instrument, da in diesen Forschungsergebnisse kodifi-

ziert werden, die innerhalb von realen soziotechnischen Systemen evaluiert werden können (Untersuchung von Verhalten und Eignung). Zusätzlich zu traditionellen, wissenschaftlichen Verbreitungskanälen für Artefakte der konstruktionsorientierten Forschung bietet Open-Source-Software zusätzliche Transparenz und – bis zu einem gewissen Grad – Reproduzierbarkeit der durchgeführten Forschung.

Referenzen

- Abrahamsson P, Oza N, Siponen M (2010) Agile software development methods: A comparative review. In: Dingsøyr, T., Dybå T, Moe NB (eds) *Agile Software Development*. Springer, pp 31–59
- Alberer G, Alberer P, Enzi T, Ernst G, Mayrhofer K, Neumann G, Rieder R, Simon B (2003) *The Learn@WU Learning Environment*. In: *Proc Wirtschaftsinformatik 2003*, Dresden
- Apel S, Kästner C (2009) An overview of feature-oriented software development. *Journal of Object Technology*, 8(5):49–84
- Allen BR, Boynton AC (1991) Information architecture: In search of efficient flexibility. *MIS quarterly*, 15(4):435–445
- Andergassen M, Guerra V, Ledermüller K, Neumann G (2012) Browser-based mobile clickers: Implementation and challenges. In: *Proc IADIS International Conference - Mobile Learning 2012*, Berlin
- Bjerknes G, Bratteteig T, Espeseth T (1991) Evolution of finished computer systems. *Scandinavian Journal of Information Systems*, 3:25–45
- Breu R, Agreiter B, Farwick M, Felderer M, Hafner M, Innerhofer-Oberperfler F (2011) Living models - Ten principles for change-driven software engineering. *Int. J. Software and Informatics*, 5(12):267–290
- Brynjolfsson E, Hitt LM, Kim HH (2011) *Strength in Numbers: How Does Data-Driven Decision-making Affect Firm Performance?* SSRN working paper, Social Science Research Network (SSRN)
- Callaù O, Robbes R, Tanter E, Röthlisberger D (2011) How (and why) developers use the dynamic features of programming languages: the case of Smalltalk. *Empirical Software Engineering*, pp 1–39
- Clements P, Northrop L (2007) *Software Product Lines: Practices and Patterns*. Addison-Wesley
- Fowler M (2010) *Domain-specific languages*. Pearson Education
- Frank U (2006) *Towards a pluralistic conception of research methods in information systems research*. ICB Research Reports 7, University Duisburg-Essen, Institute for Computer Science and Business Information Systems
- Georgakopoulos D, Hornick M, Sheth A (1995) An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and parallel Databases*, 3(2):119–153
- Germonprez M, Hovorka D, Collopy F (2007) A theory of tailorable technology design. *J. AIS*, 8(6):351–367
- Germonprez M, Hovorka D, Gal U (2011) Secondary design: A case of behavioral design science research. *J. AIS* 12(10):662-683
- Hansen HR, Neumann G (2009) *Wirtschaftsinformatik 1, Grundlagen und Anwendungen, Band 1*, 10. Auflage. Lucius & Lucius, UTB 2669, ISBN 978-3-8282-0479-9
- Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. *MIS quarterly*, 28(1):75–105
- Hoyer V, Gilles F, Janner T, Stanoevska-Slabeva K (2009) SAP research RoofTop marketplace: Putting a face on service-oriented architectures. In: *2009 World Conference on Services - I*, pp 107–114
- Kieser A, Kubicek H (1992) *Organisation*, 3. Auflage, de Gruyter, Berlin

- Malone TW, Crowston K (1994) The interdisciplinary study of coordination. *ACM Comput. Surv.* 26(1):87–119
- Malone TW, Crowston K, Lee J, Pentland B, Dellarocas C, Wyner G, Quimby J, Osborn CS, Bernstein A, Herman G, Klein M, O'Donnell E (1999) Tools for Inventing Organizations: Toward a Handbook of Organizational Processes. *Management Science*, 45(3):425–443
- March ST, Smith GF (1995) Design and natural science research on information technology. *Decision Support Systems*, 15(4):251–266
- Mernik M, Heering J, Sloane AM (2005) When and how to develop domain-specific languages. *ACM Comput. Surv.* 37(4):316-344
- Mödritscher F, Andergassen M, Neumann G (2013) Dependencies between e-learning usage patterns and learning results. In: *Proc i-KNOW 2013, 13th International Conference on Knowledge Management and Knowledge Technologies*, Graz
- Muller MJ, Kuhn S (1993) Participatory design. *Commun. ACM*, 36(6):24–28
- Neumann G, Erol S (2008) From a social wiki to a social workflow system. In: Ardegnà D, Mecella M, Yang J (eds) *Business Process Management Workshops 2008 LNBIP 17*, Milan, pp 698–708
- Orlikowski WJ, Iacono CS (2001) Research commentary: Desperately seeking the "IT" in IT Research - A call to theorizing the IT artifact. *Information Systems Research*, 12(2):121–134
- Österle H, Otto B (2010) Consortium Research: A Method for Researcher-Practitioner Collaboration in Design-Oriented IS Research. *Business & Information Systems Engineering*, 2(5):283–293
- Ousterhout JK (1998) Scripting: Higher level programming for the 21st century. *IEEE Computer*, 31(3):23–30
- Pahlke I, Beck R, Wolf M (2010) Enterprise mashup systems as platform for situational applications. *Business & Information Systems Engineering*, 2(5):305–315
- Peffer K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A design science research methodology for information systems research. *Journal of Management Information Systems*, 24:45–77
- Rank S (2002) A Reflective Architecture to Support Dynamic Software Evolution. PhD thesis, University of Durham, UK
- Sein MK, Henfridsson O, Purao S, Rossi M, Lindgren R (2011) Action design research. *MIS Quarterly*, 35(1):37–56
- Siemens G, Long P (2011) Penetrating the Fog: Analytics in Learning and Education. *EDUCAUSE Review*, 46(5):30–32
- Stahl T, Voelter M, Czarnecki K (2006) *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons
- Strembeck M, Zdun U (2009) An Approach for the Systematic Development of Domain-Specific Languages. *Softw., Pract. Exper.*, 39(15):1253–1292
- Vaishnavi VK, Kuechler W (2007) *Design Science Research Methods and Patterns: Innovating Information and Communication Technology*. Auerbach Publications
- Van der Aalst WMP (2011) *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer
- Von Hippel E (2009) Democratizing innovation: The evolving phenomenon of user innovation. *International Journal of Innovation Science*, 1(1):29–40
- Wulf V, Rohde M (1995) Towards an integrated organization and technology development. In *Proceedings of the 1st conference on Designing interactive systems: processes, practices, methods, & techniques*, ACM, pp 55-64

Wulf V (1999) Evolving cooperation when introducing groupware: a self-organization perspective. *Cybernetics & Human Knowing*, 6(2):55-74