

2 Grundlagen

In diesem Kapitel werden die theoretischen und begrifflichen Grundlagen der vorliegenden Arbeit vorgestellt. Hierzu wird in Unterkapitel 2.1 zunächst die *Systemtheorie* als theoretische Linse der Arbeit diskutiert. Nach der Einführung der relevanten Grundlagen im Bereich der *Architekturen* in Unterkapitel 2.2 werden in Unterkapitel 2.3 die zuvor eingeführten Konzepte zu dem theoretischen Bezugsrahmen der Arbeit zusammengeführt.

2.1 Systemtheorie

Wesentliche theoretische Grundlage der vorliegenden Arbeit stellt die Systemtheorie dar. Dieses Unterkapitel ist in drei Abschnitte aufgeteilt. In Abschnitt 2.1.1 werden zunächst der *allgemeine Systembegriff* sowie die drei Systemkonzepte nach Ropohl (2009) dargestellt. Nach einem kurzen Abriss der Entwicklung hin zur *modernen Systemtheorie* in Abschnitt 2.1.2 wird in Abschnitt 2.1.3 die Theorie *komplexer adaptiver Systeme* eingeführt.

2.1.1 Allgemeiner Systembegriff

Der Systembegriff geht auf das altgriechische σύστημα [sýstēma] zurück, was soviel bedeutet wie „aus mehreren Teilen zusammengesetztes und gegliedertes Ganzes“. In der wissenschaftlichen Literatur existiert eine Reihe von Definitionen für den Begriff „System“. Ein weit verbreitetes Systemverständnis, welches den Fokus auf die System-Elemente und die Beziehungen zwischen diesen legt, findet sich dabei in einigen Variationen. Van Gigch (1991) definiert ein System wie folgt:

„A system is an assembly or set of related elements.“ (Van Gigch 1991, S. 30)

In eine sehr ähnliche Richtung geht die Definition von Hall & Fagen (1969), die jedoch auch Beziehungen zwischen den Attributen (oder Eigenschaften) von Elementen bzw. Objekten berücksichtigt:

„A system is a set of objects together with relationships between the objects and between their attributes.“ (Hall & Fagen 1969, S. 31)

Der Standard ISO/IEC 15288:2008 (Systems and software engineering – System life cycle processes) betont neben miteinander in Beziehung stehenden bzw. interagierenden Elementen zudem auch den Systemzweck:

„[A system is a] combination of interacting elements organized to achieve one or more stated purposes.“ (ISO/IEC 2008, S. 6)

Die vorliegende Arbeit folgt der Definition nach Hall & Fagen (1969). Mögliche Beispiele für Systeme im Sinne des gewählten Systembegriffs sind sehr vielseitig und reichen von Wirtschafts-, Öko-, Umwelt- und Nervensystemen über Betriebssysteme bis hin zu Waffensystemen.

Ropohl (2009) diskutiert in seiner Arbeit drei Systemkonzepte, die je nach Systemverständnis der wissenschaftlichen Autoren in den Vordergrund rücken, wobei nach Ropohl (2009) alle drei Konzepte durch den Systembegriff abgedeckt werden sollten. Bei diesen Konzepten handelt es sich um das *funktionale*, das *strukturelle* sowie das *hierarchische* Systemkonzept.

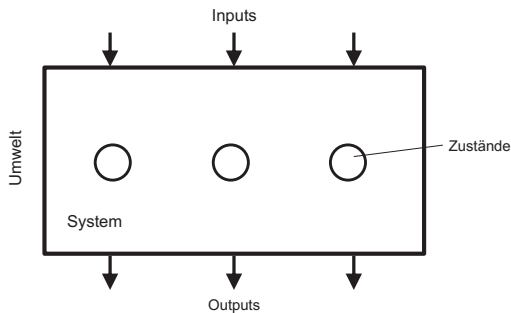


Abbildung 2: Funktionales Konzept nach Ropohl (2009)

Das funktionale Systemkonzept (vgl. Abbildung 2) abstrahiert vom inneren Aufbau des Systems und fokussiert auf das Verhalten des Systems in seiner Umwelt. Hierbei wird das System – bis auf die es beschreibenden Zustände – als „black box“ betrachtet, die bestimmte Eigenschaften aufweist. Zu diesen Eigenschaften zählen neben den bereits angeführten, das System beschreibenden Zuständen, die Eingangsgrößen (Inputs) und die Ausgangsgrößen (Outputs) des Systems. Wie das System aus den Eingangs- die entsprechenden Ausgangsgrößen generiert, wird durch das funktionale Konzept ausgeblendet.

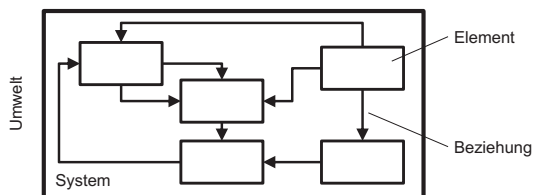


Abbildung 3: Strukturelles Konzept nach Ropohl (2009)

Bei dem strukturalen Konzept (vgl. Abbildung 3) steht der Aufbau des Systems „[...] als eine Ganzheit miteinander verknüpfter Elemente [...]“ (Ropohl 2009, S. 75) im Fokus. Unterschiedliche Kombinationen und Verknüpfungen der gegebenen Systemelemente können zu einer Vielzahl von Systemeigenschaften führen. Das strukturelle Konzept folgt daher dem Grundsatz, dass die Systemelemente nicht unabhängig von ihrem Kontext – d. h. den übrigen Systemelementen sowie der Verknüpfungen zwischen diesen – betrachtet werden sollten.

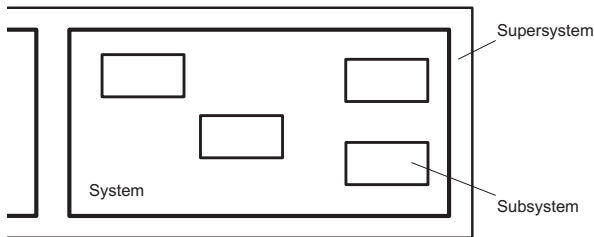


Abbildung 4: Hierarchisches Konzept nach Ropohl (2009)

Das hierarchische Konzept (vgl. Abbildung 4) betrachtet die Elemente eines Systems ebenfalls als Systeme, sogenannte Subsysteme. Nach Simon (1962) können Systeme in weitere Subsysteme untergliedert werden, bis eine unterste Ebene elementarer Subsysteme erreicht wird.⁶ Gleichzeitig kann das System aber auch als Element eines übergeordneten Systems, dem sogenannten Supersystem, verstanden werden. Aus einer systemanalytischen Perspektive unterstützt das hierarchische Konzept in zweierlei Hinsicht. Ein „Hineinzoomen“ in eine tiefere Hierarchieebene erlaubt ein besseres Verständnis und eine detailliertere Erklärung des Systems, wohingegen ein „Herauszoomen“ zu einem Erkenntnisgewinn hinsichtlich der Systembedeutung führt (Mesarovic & Macko 1969).

Es ist ersichtlich, dass sich die drei angeführten Systemkonzepte nicht ausschließen, sondern einander vielmehr komplementieren. Je nach Perspektive auf den Untersuchungsgegenstand erlauben die Systemkonzepte dem Betrachter einen unterschiedlichen Erkenntnisgewinn. Im Zuge dieser Arbeit sind primär die Perspektiven auf die Struktur der Elemente auf Systemebene – das strukturelle Konzept – sowie auf die hierarchische Betrachtung mit Blick auf die Zuordnung der Subsysteme zu den darüberliegenden Systemen – das hierarchische Konzept – im Fokus. Der in dieser Arbeit zugrunde liegende Systembegriff nach Hall & Fagen (1969) erwähnt die angeführten Systemkonzepte in ihrer Ganzheit zwar nicht explizit, ist jedoch auf diese anwendbar.

2.1.2 Wurzeln der modernen Systemtheorie

Nachdem im vorangegangenen Abschnitt der Systembegriff sowie die verschiedenen Systemkonzepte eingeführt wurden, gibt dieser Abschnitt einen kurzen Abriss über die Entwicklung der noch relativ jungen Disziplin der Systemwissenschaft hin zur modernen Systemtheorie. Die Ursprünge letzterer sind im Wesentlichen in der allgemeinen Systemtheorie (Bertalanffy 1948; Bertalanffy 1968), der Kybernetik (Wiener 1961) sowie der Chaostheorie zu finden.

Der Biologe Ludwig von Bertalanffy beobachtete seinerzeit eine immer weiter fortschreitende Spezialisierung der einzelnen Forschungsdisziplinen, aus denen sich jeweils wiederum eine

⁶ Vgl. hierzu auch den von Simon (1962) verwendeten Begriff hierarchischer Systeme: „By a hierarchic system, or hierarchy, I mean a system that is composed of interrelated subsystems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem. In most systems in nature, it is somewhat arbitrary as to where we leave off the partitioning, and what subsystems we take as elementary“ (Simon 1962, S. 468).

Vielzahl von Subdisziplinen herausbildete. Jede dieser Disziplinen und Subdisziplinen entwickelte sich zu einem eigenen „privaten Universum“ mit eigenen Methoden und Termini, was es zunehmend erschwerte, ein „*Wort von einem Kokon in einen anderen zu bekommen*“ (Bertalanffy 1968, S. 30). Gleichzeitig beobachtete von Bertalanffy jedoch, dass sich in verschiedenen Forschungsgebieten vergleichbare Problemstellungen und Konzepte herausgebildet hatten. In einer allgemeinen Systemtheorie sah von Bertalanffy eine Möglichkeit, die Zusammenarbeit einzelner Forschungsgebiete zu fördern und theoretische Konzepte zu identifizieren, die nicht auf Einzelwissenschaften beschränkt sind. Diese Konzepte sollten demnach auf diverse Systeme anwendbar sein. Ludwig von Bertalanffy diskutierte in seinen Arbeiten Konzepte wie geschlossene und offene Systeme, Selbstregulation und Gleichgewicht (Bertalanffy 1968). Das Interesse, ein besseres Verständnis für gemeinsame, grundlegende Konzepte verschiedener Disziplinen zu entwickeln, teilte von Bertalanffy mit einer Vielzahl von Forschern aus den unterschiedlichsten Forschungsgebieten. Im Rahmen von Symposien kam es zu einem Austausch von Wissenschaftlern aus verschiedensten Bereichen wie u. a. dem Ingenieurwesen, der Mathematik, der Volkswirtschaftslehre, der Biologie oder der Philosophie (Mesarovic 1964).

Auch der Mathematiker Norbert Wiener und der Psychiater William Ross Ashby trugen mit ihren Arbeiten zur Kybernetik – der zweiten Strömung hin zur modernen Systemtheorie – zu einem disziplinübergreifenden Erkenntnisgewinn bei (Ashby 1956; Wiener 1961). In seinem Buch „*Cybernetics or Control and Communication in the Animal and the Machine*“ zeigte Wiener u. a., dass sowohl in künstlichen Systemen als auch in lebendigen Organismen vergleichbare Regelungs- und Steuerungsvorgänge zu beobachten sind (Wiener 1961).

Ein parallel dazu entstandenes, verwandtes Gebiet ist das der Chaostheorie, die Phänomene einer Reihe nichtlinearer dynamischer Systeme beschreibt. Das Verhalten solcher Systeme wird als chaotisch bezeichnet, wenn dieses empfindlich auf eine Variation der Anfangsbedingungen reagiert, obwohl sich die ihm zugrunde liegenden Gesetzmäßigkeiten deterministisch verhalten. Ein Beispiel für ein solches, sensibel auf Anfangswerte reagierendes System ist das Wetter. Edward Lorenz etablierte an diesem Beispiel den Begriff des „Schmetterlingseffektes“⁷ und legte einen wichtigen Grundstein für die Entstehung der Chaostheorie (Gleick 1988; Lorenz 2000). Ein Beispiel für chaotisches Verhalten einfacher nichtlinearer Gleichungen ist die sogenannte logistische Gleichung⁸ (May 1976; Verhulst 1838).

Die noch relativ junge Theorie komplexer adaptiver Systeme wurde stark von Arbeiten des Mitte der 90er Jahren gegründeten Santa Fe Institutes (New Mexico) geprägt (vgl. hierzu Holland 1995; Holland 1998; Sneppen et al. 1995). Da diese Theorie wesentlich für das Ver-

⁷ Edward Lorenz diskutierte im Rahmen des 139. Treffens der American Association for the Advancement of Science den potenziellen Einfluss eines Flügelschlags eines Schmetterlings auf die Entstehung eines Tornados, um die Schwierigkeit in der Vorhersage von Wetterphänomenen zu illustrieren (Lorenz 2000).

⁸ Verhulst führte die logistische Gleichung 1838 als mathematisches Modell zur Darstellung der Entwicklung einer Population ein. Die logistische Gleichung der Form $x_{n+1} = rx_n - rx_n^2$ erzeugt bei Variation des konstanten Parameters r im Bereich $[2, 4]$ Folgen in der Umgebung einer zunehmenden Anzahl von Häufungspunkten. Mit steigendem Wert für r nimmt die Anzahl der Häufungspunkte zu. Gleichzeitig verkürzen sich die Intervalle mit gleicher Anzahl von Häufungspunkten bis die Folge zu chaotischem Verhalten übergeht.

ständnis der vorliegenden Arbeit und deren theoretischen Beitrag im Bereich der Systemtheorie ist, führt der folgende Abschnitt näher in die Annahmen und Charakteristiken, die dieser Theorie zugrunde liegen, ein.

2.1.3 Komplexe adaptive Systeme

Das Forschungsgebiet der komplexen adaptiven Systeme – im weiteren Verlauf der Arbeit abgekürzt mit „CAS“ (Complex Adaptive System) – stellt eine vergleichsweise neue Weise dar, über Systeme zu denken, die sich von den bestehenden Theorien deutlich abgrenzt. Entgegen mechanistischer Theorien, die eine Form zentraler Kontrollstruktur vorsehen, basiert die Theorie der CAS auf der Idee, dass Ordnung durch die Interaktion der Systemelemente emergiert (Benbya & McKelvey 2006). Gegenüber der Chaostheorie, die sich auf nicht voraussagbares Verhalten deterministischer Gleichungen fokussiert, beschäftigt sich die Theorie der CAS mit Ordnung und der Frage, wie diese Ordnung entsteht (Benbya & McKelvey 2006). Ein wesentlicher Fokus der CAS liegt dabei auf dem Zusammenspiel der Systeme mit ihrer Umwelt und der Koevolution, also des sich miteinander Weiterentwickelns, der Systeme und ihrer Umwelt (Anderson 1999; Choi et al. 2001).

Holland (1995) definiert CAS als Systeme, die aus einer großen Anzahl *interagierender, heterogener*⁹ „aktiver Elemente“ – sogenannter *Agenten* – bestehen. Das Verhalten der Agenten wird durch eine Menge an *Regeln* bestimmt. Diese Regeln haben typischerweise die Form eines Reiz-Reaktionsmusters: „wenn Reiz s , dann Reaktion r “ (vgl. Abbildung 5).

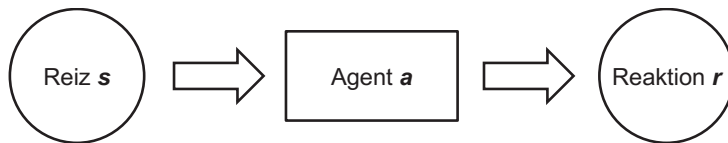


Abbildung 5: Reiz-Reaktions-Schema

Die Menge der Reiz-Reaktions-Regeln eines Agenten können durch die Reize, die der Agent empfangen kann, sowie die Reaktionen, die der Agent auf die Reize hin zeigt, beschrieben werden. Die Sequenzen, die aus den Reiz-Reaktions-Regeln gebildet werden können, beschreiben die möglichen Verhaltensmuster¹⁰ des Agenten. Basierend auf Erfahrungen, die Agenten sammeln, können sich die Agenten an ihre Umgebung anpassen, indem sie die den Verhaltensmustern zugrunde liegenden Regeln ändern. Diese Fähigkeit der Agenten wird auch als *Adaption und Lernen* bezeichnet (Holland 1995).

Das Verhalten des Gesamtsystems wird weder extern vorgegeben, noch wird es durch einen einzelnen Agenten bestimmt. Systemverhalten entsteht aus der Interaktion aller Agenten

⁹ Unterschiedlich in Gestalt und Fähigkeiten (vgl. Holland 1995, S. 6)

¹⁰ Kim & Kaplan (2006) verstehen diese Verhaltensmuster als kontextsensitive Strategien: „Actors, or populations of actors, interact with their environment and other actors within neighbourhoods, and employ a variety of context bound strategies that may be planned and purposeful or conditioned and reactive.“ (Kim & Kaplan 2006, S. 3)

(Choi et al. 2001). CAS durchlaufen dabei einen Prozess der *Selbstorganisation*, durch den neues Systemverhalten *emergiert*.

Die Agenten sind dabei nicht alle unmittelbar miteinander „verknüpft“, sondern interagieren nur mit einer Reihe von Agenten in ihrem Umfeld. Sowohl die Muster, nach denen die Agenten untereinander verknüpft sind, als auch die „Stärke“ der Verknüpfung sind dabei nicht statisch vorgegeben, sondern können sich im Zeitverlauf ändern. Durch regen Austausch und positive Erfahrungen bei der Interaktion (z. B. Wertschätzung der Zusammenarbeit) können sich die Verbindungen verstärken, sich nach einer einmaligen Interaktion jedoch auch wieder auflösen. Gleichmaßen können durch den Bedarf eines Austausches neue Verbindungen entstehen. Hierdurch ist eine Dynamik in der Struktur des Systems (im Sinne einer Netzwerktopologie) zu beobachten (Merali 2006). An dieser Stelle kommt auch das Konzept der beschränkten Rationalität („bounded rationality“; vgl. hierzu auch March & Simon (1958)) zum Tragen. Kein Agent hat vollständige Informationen über den Zustand des gesamten Systems zu einem bestimmten Zeitpunkt. Agenten agieren auf Basis einer beschränkten Informationsmenge über einen unmittelbaren Ausschnitt ihrer aktuell vorzufindenden Umgebung inklusive der Agenten im direkten Umfeld (vgl. Kim & Kaplan 2006). Demnach emergiert der Zustand des gesamten Systems stets aus dem lokalen Verhalten der Agenten, welches wiederum aus deren Erfahrungen und den mit diesen verbundenen Verhaltensmustern resultiert (Merali 2006). Evolution und Lerneffekte einzelner Agenten haben dabei Einfluss auf das Verhalten, die Evolution und die Lerneffekte anderer, mit diesen interagierenden Agenten. Dies führt zur sogenannten *Koevolution* der Agenten sowie der Koevolution der Agenten und des Systems (Anderson 1999). Die wesentlichen Charakteristiken der CAS sind in Tabelle 1 zusammengefasst.

Beispiele für CAS sind u. a. das zentrale Nervensystem (Agenten: Neuronen), das Immunsystem (Agenten: Antikörper), ein Unternehmen (Agenten: Mitarbeiter) eine Ökonomie (Agenten: Unternehmen) oder auch ein Ökosystem (Agenten: Lebewesen). Die Zeitdauer bis zu einer Änderung des CAS – basierend auf den Adaptionen- und Lernprozessen – bewegt sich im Falle der Nervensysteme im Bereich von Sekunden bis Stunden. Bei Immunsystemen liegt diese bei Stunden bis Tagen. Im Falle von Unternehmen bei Monaten bis Jahren (Holland 1995).

Neben den hier angeführten Beispielen aus der Biologie und der Ökonomie wird das Konzept CAS auch zunehmend auf den IS-Kontext angewandt. So untersucht Van Aardt (2004) die Entwicklung von Open Source Software als CAS und Hanseth & Lyytinen (2010) schlagen in ihrer Arbeit eine auf CAS basierende Designtheorie für die dynamische Komplexität in der Gestalt von Informationsinfrastrukturen am Beispiel des Internets vor. Benbya & McKelvey (2006) gehen einen Schritt weiter, indem sie jedes Informationssystem als CAS betrachten. Allen & Varga (2006) führen hingegen an, dass IT-Systeme – als Hard- und Software ein wesentlicher Teil der Informationssysteme – nicht die agentenhaften Eigenschaften eines CAS aufzeigen und sich somit auch nicht – im Sinne einer Koevolution – an sich ändernde Realitäten anpassen. Auf diesen Punkt geht die vorliegende Arbeit in Unterabschnitt 3.1.2.5 weiter ein.

Tabelle 1: Charakteristiken komplexer adaptiver Systeme

Charakteristik	Beschreibung	Literatur
Heterogene Agenten	CAS bestehen aus heterogenen Agenten, die sowohl untereinander als auch mit ihrer Umgebung interagieren und ihr Verhalten auf Grundlage vorheriger Erfahrungen anpassen können. Aufgrund der Möglichkeit kontinuierlicher Anpassung der lokal agierenden Agenten innerhalb des Systems, kann aus dem Verhalten der Agenten nicht auf das Verhalten des Gesamtsystems geschlossen werden.	Benbya & McKelvey (2006) Holland (1995)
Interaktion	Das Verhalten der Agenten kann sowohl durch die anderen Agenten als auch das System beeinflusst werden. Die Agenten agieren entsprechend ihrer Beziehungen innerhalb des Systems und ihrer Regelmenge.	Benbya & McKelvey (2006) Holland (1995)
Adaption und Lernen	Das Verhalten der Agenten wird vorgegeben durch Schemata. Ein Schema ist als eine Abfolge von Regeln, die aufgrund des durch den Agenten wahrgenommenen Zustandes der Umwelt anzuwenden ist, zu verstehen. Agenten können über geteilte als auch individuelle Schemata verfügen. Das Bilden neuer oder Anpassen bestehender Schemata aufgrund von Erfahrungen (z. B. einer sich ändernden Umweltsituation) oder der Variation bestehender Regeln wird als Adaption und Lernen bezeichnet.	Anderson (1999) Holland (1995)
Selbstorganisation und Emergenz	(Neues) Systemverhalten entsteht aus der Interaktion der Agenten: Entgegen der Kontrolltheorien aus der Kybernetik bestimmt weder ein einzelner Agent das Verhalten des Systems, noch werden die neu entstehenden Strukturen, Eigenschaften und Verhaltensweisen des Systems extern vorgegeben. Vielmehr durchlaufen CAS einen Prozess der Selbstorganisation in dessen Rahmen neues Systemverhalten durch die interagierenden Agenten entsteht (emergiert).	Anderson (1999) Benbya & McKelvey (2006) Choi et al. (2001) Holland (1998)
Koevolution	„Ein CAS und seine Umwelt interagieren und bilden dabei dynamische emergierende Realitäten“ (Choi et al. 2001, S. 356): Da Agenten – in Anlehnung an das Konzept der beschränkten Rationalität („bounded rationality“, vgl. hierzu auch March & Simon (1958)) – nicht die vollständigen Konsequenzen ihres Verhaltens für das Gesamtsystem beurteilen können, optimieren sie ihr Verhalten hinsichtlich ihrer eigenen Ziele, nicht jedoch hinsichtlich der des Gesamtsystems. Evolution und Lerneffekte einzelner Agenten haben Einfluss auf das Verhalten, die Evolution und die Lerneffekte anderer, mit diesen interagierenden Agenten. Dies führt zur sogenannten Koevolution der Agenten. Die gleiche Logik gilt für die Koevolution von interagierenden Mengen von Agenten sowie die des gesamten CAS mit dessen Umwelt.	Anderson (1999) Choi et al. (2001) March & Simon (1958) Merali (2006)

2.2 Architekturen

In diesem Unterkapitel wird zunächst der *allgemeine Architekturbegriff* eingeführt (Abschnitt 2.2.1) und darauf folgend auf den *Architekturbegriff im Kontext der Informationssysteme* eingegangen (Abschnitt 2.2.2). Im Anschluss daran werden in den Abschnitten 2.2.3 und 2.2.4 die Konzepte der *Unternehmensarchitektur* sowie der *IT-Architektur* diskutiert und voneinander abgegrenzt. Das *Architekturmanagement* – als eine Managementdisziplin in Unternehmen – wird in Abschnitt 2.2.5 dargestellt, bevor in Abschnitt 2.2.6 die *Rolle des Architekten* diskutiert wird.

2.2.1 Allgemeiner Architekturbegriff

Etymologisch geht der Begriff „Architektur“ auf das aus dem Lateinischen stammende Wort „*architectura*“ zurück. Dieses ist aus den griechischen Wörtern „*ἀρχή*“ (gesprochen [arché]; Bedeutung: „Anfang“, „Ursprung“, „Grundlage“ oder „das Erste“) und „*τέχνη*“ (gesprochen [techné]; Bedeutung: „Kunst“ oder „Handwerk“) zusammengesetzt. Es lässt sich demnach als „das Erste Handwerk“ oder „die Erste Kunst“ übersetzen (Pfeifer 1997; Seebold & Kluge 2011). Die heutige Bedeutung des Begriffs Architektur hängt von dem Kontext sowie dem Aufgabenfeld des „Architekten“¹¹ ab.

Bei Betrachtung des aus zehn Büchern bestehenden Werkes des römischen Architekten und Ingenieurs Marcus Vitruvius Pollio aus dem 1. Jahrhundert v. Chr. wird bereits der starke Bezug des klassischen Architekturbegriffs zum Städtebau sowie zur Gestaltung von Bauwerken wie Tempeln und sonstigen öffentlichen und privaten Gebäuden deutlich (Vitruvius Pollio 2004). In der Literatur ab der Neuzeit findet der Architekturbegriff keine einheitliche Verwendung. Ruskin (1865) versteht Architektur als die Kunst der Erbauung von Bauwerken:

“Architecture is the art which so disposes and adorns the edifices raised by man for whatsoever uses, that the sight of them contribute to his mental health, power and pleasure.” (Ruskin 1865, S. 7)

Browne (1658) legt bei seiner Verwendung des Architekturbegriffs den Fokus auf den Erstellungsprozess von Gebäuden bzw. Städten. Eine weitere Verwendung des Architekturbegriffs zeigt sich in der Bezeichnung der Gesamtheit von Bauten eines Stiles, eines Volkes bzw. einer Epoche. John Ruskin nennt in seinem Werk „*Stones of Venice*“ u. a. den byzantinischen, den romanischen und den gotischen Baustil (Ruskin 1851).

Mittlerweile hat der Architekturbegriff Einzug in die verschiedensten wissenschaftlichen Domänen gefunden und findet dort u. a. zur Beschreibung einer bestimmten Konstruktion oder Struktur Anwendung. So wird z. B. in der Biologie und der Chemie bei der Untersuchung molekularer Strukturen häufig auch von einer molekularen Architektur gesprochen (vgl. z.B. Ferreira et al. 2004; Pauling 1946; Woese et al. 1990).

¹¹ Der Begriff „Architekt“, lat. *architectus*, entstand aus dem altgriechischen *αρχιτέκτων* (gesprochen [architék-ton]). Letzteres setzt sich zusammen aus „*αρχι*“ (gesprochen [archi]) – was so viel bedeutet wie „erster“, „oberster“ oder „höchster“ – sowie „*τέκτων*“ (gesprochen [téktion]) – „Handwerker, Zimmermann“.

2.2.2 Der Architekturbegriff im Kontext von Informationssystemen

Für die Errichtung von Bauwerken werden bestimmte Elemente, wie unterschiedliche Baustoffe und Materialien, Werkzeuge, Baupläne, Skizzen und Vorgehensweisen benötigt. Analog zur klassischen Baukunst ist auch die Architektur von Informationssystemen¹² charakterisiert durch eine Vielzahl von miteinander in Beziehung stehenden Elementen und den dazugehörigen Prozesslandkarten und Dokumentationen (Matthes 2011). So überrascht es nicht, dass der Architekturbegriff auf Strukturen in den Feldern der Informatik und der Wirtschaftsinformatik übertragen wurde (Aier et al. 2008; Espinosa et al. 2011; Kruchten et al. 2006). Auch Begriffe aus der klassischen Architektur – wie dem Städtebau – haben mittlerweile Einzug in die Sprache der Architektur von Informationssystemen gefunden. Ein Beispiel hierfür ist der Begriff der Bebauungsplanung, der im Kontext der Informationssysteme zur Dokumentation der bestehenden (As-Is) sowie der geplanten (To-Be) IT-Landschaft Verwendung findet (Aier et al. 2012; Niemann 2005; Saat et al. 2009).

Wie bei der Einführung des allgemeinen Architekturbegriffs deutlich wurde, kann dieser unterschiedliche Perspektiven umfassen. Schmidt (2009) unterscheidet in seiner Arbeit drei Dimensionen des Architekturbegriffs:

1. Die ontologische Begriffsdimension ist auf das Objekt selbst fokussiert (z. B. ein Gebäude oder die grundlegende Struktur eines Systems).
2. Die pragmatische Begriffsdimension ist auf den (Erstellungs-)Prozess in Form von Handlungsvorschriften oder Prinzipien fokussiert. (z. B. ein Bauplan eines Hauses oder eine Menge an Prinzipien zur Entwicklung bzw. Weiterentwicklung eines Systems)
3. Die deskriptive Begriffsdimension ist auf die Beschreibung eines Objektes anhand von Modellen fokussiert¹³. Im Sinne der Modelltheorie handelt es sich hierbei um zweckorientierte Abbildungen der Struktur eines Systems (Schmidt 2009).

Die Internationale Organisation für Normung (ISO), die Internationale Elektrotechnische Kommission (IEC) und das Institute of Electrical and Electronics Engineers (IEEE) führen in dem gemeinsam veröffentlichten Standard ISO/IEC/IEEE 42010:2011¹⁴ Definitionen der für die Beschreibung der Architektur von Systemen relevanten Begriffe an (ISO/IEC/IEEE 2011). Architektur wird hier definiert als:

¹² Krcmar (2009) folgend werden "Informationssysteme" als soziotechnische Systeme verstanden, bestehend aus menschlichen sowie maschinellen Komponenten. Die maschinellen Komponenten teilen sich wiederum in Anwendungs- und Hardware-Komponenten auf.

¹³ Das im Rahmen der vorliegenden Arbeit verwendete Systemkonzept ist eng verknüpft mit der Modelltheorie. So werden reale Sachverhalte (wie u. a. die in Abschnitt 2.1.3 angeführten Beispiele) meist unter Zuhilfenahme von Konzepten der Modelltheorie beschrieben. Nach der allgemeinen Modelltheorie (Stachowiak 1973) ist unter einem Modell eine vereinfachte (Verkürzungsmerkmal) Darstellung (Abbildungsmerkmal) eines Originals zu verstehen. Ein Modell erfüllt dabei eine Ersetzungsfunktion des Originals, wobei sich die Gestalt des Modells auf die durch den jeweiligen Ersteller als wesentlich erkannten Elemente beschränkt und zu einem bestimmten Zeitpunkt sowie für einen definierten Zweck erstellt wird (pragmatisches Merkmal).

¹⁴ Der ISO/IEC/IEEE 42010:2011 "Systems and Software engineering – Architecture description" ersetzt den vielzitierten IEEE Std. 1471-2000 "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems" (IEEE Architecture Working Group 2000), welcher durch die ISO und die IEC auch als ISO/IEC 42010:2007 geführt wurde.

„[...] *fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution*“ (ISO/IEC/IEEE 2011, S. 2)

Nach dieser Definition besteht eine Architektur aus den Konzepten und Eigenschaften eines Systems, die sich sowohl am realen System selbst zeigen als auch an den (verschriftlichten) Prinzipien bezüglich der Ausgestaltung und Weiterentwicklung des Systems. Damit berücksichtigt diese Definition die ontologische und die pragmatische Begriffsdimension. Die deskriptive Perspektive wird im ISO/IEC/IEEE (2011) durch die Architekturdokumentation aufgegriffen. Die sogenannte „architecture description“ wird dabei definiert als:

„*work product used to express an architecture*“ (ISO/IEC/IEEE 2011, S. 2)

Die angeführten Arbeitsergebnisse umfassen u. a. die Identifikation der Systemakteure samt deren Interessen und Belangen, die Definition der für die jeweilige Architektur zugrunde liegenden Architekturperspektiven (sog. „architecture viewpoints“), Architekturmodelle für die verwendeten Architekturperspektiven, aber auch getroffene Architekturentscheidungen, deren Begründungen sowie die durch diese entstehenden Implikationen für das System und dessen Umfeld.

Die vorliegende Arbeit verwendet den Architekturbegriff im Sinne der ontologischen Begriffsdimension. In Anlehnung an den ISO/IEC/IEEE (2011) wird *Architektur* im Kontext von Informationssystemen aus einer systemtheoretischen Perspektive wie folgt definiert: *Eine Architektur ist die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht.*

2.2.3 Unternehmensarchitektur

Der zuvor eingeführte Architekturbegriff im Kontext von Informationssystemen unterteilt sich in der wissenschaftlichen Literatur – abhängig von dem betrachteten System – in zwei Forschungsströme. Im Forschungsfeld der Kerninformatik liegt der Fokus auf den Software-systemen und damit der Softwarearchitektur. Betrachtungsgegenstand sind hierbei die Struktur und die Organisation von Software-Komponenten und deren Schnittstellen (Kruchten et al. 2006; Parnas 1972). In der Wissenschaftsdisziplin der Wirtschaftsinformatik – die ihre Wurzeln sowohl in der Informatik als auch in den Wirtschaftswissenschaften hat – bzw. dem angelsächsischen Pendant¹⁵ der Information Systems wird eine ganzheitlichere Perspektive auf die sog. Unternehmensarchitektur (engl. „Enterprise Architecture“) angenommen. Der ganzheitliche Charakter zeigt sich anhand der über die reine Softwarearchitektur hinausgehenden Betrachtung der organisationalen und der technischen Ebene (vgl. Tabelle 2).

¹⁵ Frank et al. (2008) zeigen in ihrer Arbeit bestehende Unterschiede in den Forschungszielen sowie –methoden auf. Herrscht in der Wirtschaftsinformatik (WI) ein eher konstruktiver Ansatz vor, haben sich in der angelsächsischen Information Systems (IS) Community in erster Linie Ansätze behavioristischer Sozialforschung etabliert. In einer aktuellen Untersuchung der methodischen Schwerpunkte in der WI kommen Schreiner et al. (2015) zu dem Ergebnis, dass konstruktive Methoden in der WI weiterhin leicht überwiegen, die WI sich jedoch zunehmend den verhaltenswissenschaftlichen Methoden öffnet. Gleichzeitig etablieren sich konstruktive Ansätze in der IS-Forschung (Gregor & Hevner 2013).

Der Begriff der Unternehmensarchitektur ist sowohl in der Praxis als auch der Forschung weit verbreitet (vgl. z.B. Ahlemann et al. 2012; Aier et al. 2008; Ross et al. 2006), wird allerdings nicht einheitlich verwendet. Aier et al. (2008) diskutieren in ihrer Arbeit Publikationen mit Bezug zu Unternehmensarchitekturen und stellen dabei das unterschiedliche Verständnis des jeweiligen Gestaltungsgegenstandes der Unternehmensarchitektur sowie die verschiedenen Nutzungsformen in Literatur und Praxis heraus.

Der wesentliche Gestaltungsgegenstand etablierter Unternehmensarchitektur-Rahmenwerke¹⁶ wie TOGAF (The Open Group Architecture Framework; The Open Group 2011) oder EAP (Enterprise Architecture Planning; Spewak & Hill 1993) sind die vier Architekturebenen Geschäftsprozesse, Daten, Anwendungen und Infrastruktur (vgl. Tabelle 2).

Tabelle 2: Ebenen einer Unternehmensarchitektur

Architekturebene	Beschreibung	Quelle	Beispiel
Geschäftsprozess-architektur (GPA)	Ausgestaltung der Geschäftsprozesse, bestehend aus einer Menge von Aktivitäten, Aufgaben und Services	Ross et al. (2006), S. 48 Pulkkinen (2006), S. 2	Kreditprozess in einer Bank
Datenarchitektur (DA)	Struktur der logischen und physischen Datenbestände	The Open Group (2011), S. 29	Syntax und Semantik von Kundeninformationen in der Datenbank
Anwendungs-architektur (AA)	Struktur der Anwendungen und deren Schnittstellen; Anwendungen verwenden Daten und werden durch Infrastrukturkomponenten unterstützt.	Ross et al. (2006), S. 48 The Open Group (2011), S. 23	Fachliche Anwendungen und deren Schnittstellen
Infrastruktur-architektur (IA)	Eine Sammlung von Technologie-Komponenten (Hardware und/oder Software) die Services anbieten, um Anwendungen zu unterstützen	Pulkkinen (2006), S. 2 The Open Group (2011), S. 24, 34	Server (Hardware oder Software), Betriebssysteme, Datenbank-managementsysteme

Wie Tabelle 3 zeigt, wird in der wissenschaftlichen Literatur eine vergleichbare Aufteilung vorgenommen. Die in den Arbeiten von Richardson et al. (1990) und Iyer & Gottlieb (2004) verwendeten Begriffsdefinitionen haben einen eher technischen Fokus, was sich daran zeigt, dass die Geschäftsprozesse keine Berücksichtigung finden. Die weiteren in Tabelle 3 aufgeführten Definitionen beziehen die Geschäftsprozesse mit ein und ähneln sich im Kern grundsätzlich.

¹⁶ Die vorliegende Arbeit verwendet die Definition eines Unternehmensarchitektur-Rahmenwerkes (engl. "Architecture Framework") nach The Open Group (2011): "An architecture framework is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks." (The Open Group 2011, S. 7)

Tabelle 3: Ausgewählte Definitionen des Begriffs Unternehmensarchitektur

Quelle	Definition	Berücksichtigte Perspektiven
Richardson et al. (1990), S. 386	“[Enterprise Architecture] defines and interrelates data, hardware, software, and communications resources, as well as the supporting organization required to maintain the overall physical structure required by the architecture.”	▪ Daten ▪ Hardware ▪ Anwendungen ▪ Beziehungen ▪ Organisation für den Betrieb und die Wartung der Infrastruktur
Iyer & Gottlieb (2004), S. 587	“The decomposition of the enterprise into manageable parts, the definition of those parts, and the orchestration of the interaction among those parts constitutes the enterprise architecture.”	▪ Dekomposition in handhabbare Abschnitte ▪ Interaktionen
Ross et al. (2006), S. 9	“Enterprise Architecture is the organizing logic for business processes and IT-infrastructure, reflecting the integration and standardization requirements of the company’s operating model. The enterprise architecture provides a long-term view of a company’s processes, systems, and technologies so that individual projects can built capabilities – not just fulfill immediate needs.”	▪ Geschäftsprozesse ▪ Systeme (Informationen + Anwendungen) ▪ IT-Infrastruktur ▪ Potenziale
Jonkers et al. (2006), S. 64	“Architecture at the level of an entire organisation is commonly referred to as “enterprise architecture” (EA). It is a coherent whole of principles, methods and models that are used in the design and realisation of the enterprise’s organizational structure, business processes, information systems, and infrastructure. EA captures the essentials of the business, IT and its evolution.”	▪ Prinzipien, Methoden und Modelle ▪ Organisationsstruktur ▪ Geschäftsprozesse ▪ Informationssysteme ▪ Infrastruktur
Tamm et al. (2011), S. 141	“Enterprise architecture (EA) is the definition and representation of a high-level view of an enterprise’s business processes and IT systems, their interrelationships, and the extent to which these processes and systems are shared by different parts of the enterprise.”	▪ Geschäftsprozesse ▪ IT-Systeme ▪ Beziehungen
Schmidt & Buxmann (2011), S. 169	“The EIS [Enterprise Information System] can be defined as a firm’s information processing subsystem consisting of all information processing tasks, the applications used to perform these tasks as well as all the technical components necessary to run the applications. Just like with any other system, the fundamental organization of the EIS (as embodied in its components, their relationships to each other and to the environment) may be referred to as the system’s architecture.[...] In accordance with industry practices, we use the term EA for this.”	▪ Informationsverarbeitende Aufgaben (Daten und Prozesse) ▪ Anwendungen ▪ Technische Komponenten ▪ Beziehungen
Bradley et al. (2012), S. 98	“Broadly defined, EA is the organising logic for an organisation’s IT infrastructure and business processes.”	▪ Geschäftsprozesse ▪ IT-Infrastruktur

Obleich nicht alle Definitionen die Ebenen der Daten-, Anwendungs- und Infrastrukturarchitektur explizit aufführen – z. B. fassen Tamm et al. (2011) und Bradley et al. (2012) diese Ebenen als „IT-Systeme“ bzw. „IT-Infrastruktur“ zusammen – haben die Autoren ein vergleichbares Verständnis.

Erwähnt der Großteil der Autoren zudem diverse Ausprägungen von Beziehungen zwischen den Elementen der Architekturebenen, so führen Schmidt & Buxmann (2011) diese Beziehungen zwischen den Architekturebenen explizit an:

“[...] information processing tasks, the applications used to perform these tasks as well as all the technical components necessary to run the applications [...]” (Schmidt & Buxmann 2011, S. 169)

Die vorliegende Arbeit folgt einem holistischen Verständnis einer Unternehmensarchitektur und bezieht die Ebenen der Geschäftsprozess-, Daten-, Anwendungs- und Infrastrukturarchitektur sowie deren Beziehungen zueinander ein (vgl. Jonkers et al. 2006; Ross et al. 2006; Schmidt & Buxmann 2011).

2.2.4 IT-Architektur

Der Begriff Unternehmensarchitektur (oder Enterprise Architecture) wird in der Praxis häufig nur einer Teilmenge der o. a. vier Architekturebenen zugeordnet (Ross et al. 2006, S. 48). Ein Beispiel hierfür ist die bereits im vorherigen Abschnitt 2.2.3 angeführte Definition von Richardson et al. (1990), welche die Teilmenge der Daten-, Anwendungs- und Infrastrukturarchitektur einbezieht, die Geschäftsprozessarchitektur jedoch außen vor lässt. Da dieser von Richardson et al. (1990) gewählte IT-Fokus in der Literatur und Praxis häufig Verwendung findet und auch für die weitere Arbeit eine wichtige Rolle spielt, wird der Begriff der IT-Architektur im Folgenden eingeführt und vom Begriff der Unternehmensarchitektur abgegrenzt.

Jeanne W. Ross führt in ihrer Arbeit „Creating a Strategic IT Architecture Competency: Learning in Stages“ die folgende Definition für eine IT-Architektur an:

“Thus, at the enterprise level, an IT architecture is the organizing logic for applications, data and infrastructure technologies, as captured in a set of policies and technical choices, intended to enable the firm’s business strategy. Accordingly, the enterprise architecture implies certain IT capabilities. These capabilities are the objectives of the IT architecture, specifying what the architecture enables the business to do.” (Ross 2003, S. 31)

Eine IT-Architektur umfasst nach dieser Definition mit der Daten-, Anwendungs- und Infrastrukturarchitektur die gleichen Gestaltungsobjekte wie bereits von Richardson et al. (1990) angeführt. Gleichzeitig stellt Ross (2003) den Zusammenhang der IT-Architektur zur Geschäfts-Perspektive her: Die IT-Architektur muss zum einen die IT-Anforderungen – bedingt durch die (geplante) Ausgestaltung der Geschäftsprozessarchitektur – erfüllen, bildet zum anderen jedoch auch IT-Potenziale als Basis für die weitere Entwicklung der Geschäftsberei-

che heraus. Abbildung 6 fasst die Abgrenzung der IT-Architektur von der Unternehmensarchitektur sowie deren Beziehung zueinander zusammen.

Wie bereits in Abschnitt 2.2.2 angeführt, verwendet diese Arbeit – in Anlehnung an den ISO/IEC/IEEE (2011) – eine systemtheoretische Perspektive auf eine Architektur im Kontext von Informationssystemen. Diese Perspektive erlaubt eine Unterteilung eines hierarchischen Systems (Simon 1962) in eine Menge miteinander verbundener Subsysteme (vgl. hierzu auch das in Abschnitt 2.1.1 eingeführte hierarchische Konzept nach Ropohl (2009)). Für die IT-Architektur entspricht dies den Subsystemen „Datenarchitektur“, „Anwendungsarchitektur“ und „Infrastrukturarchitektur“ sowie zusätzlich der „Geschäftsprozessarchitektur“ im Falle der Unternehmensarchitektur (vgl. Abbildung 6).

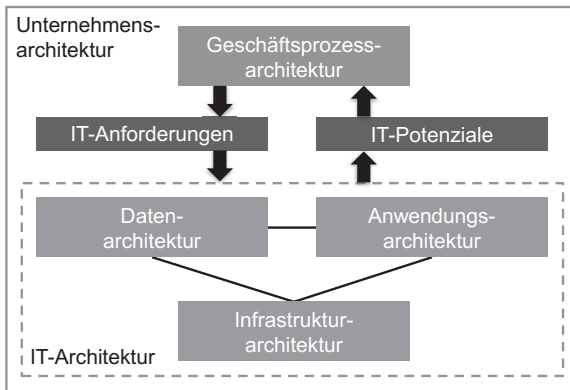


Abbildung 6: Abgrenzung Unternehmens-/IT-Architektur nach Schütz et al. (2013a)

Da die Komponenten und Beziehungen dieser Subsysteme unterschiedliche Eigenschaften aufweisen (vgl. Tabelle 2), erlaubt eine solche Unterteilung in die genannten Subsysteme eine separate Betrachtung, während die wechselseitigen Beziehungen der Subsysteme untereinander weiterhin berücksichtigt werden können.

2.2.5 Architekturmanagement

Das Architekturmanagement (engl. „Enterprise Architecture Management“ (EAM)) ist eine Management-Disziplin (Ahlemann et al. 2012), welche die Prozesse, Methoden, Werkzeuge, Verantwortlichkeiten und Standards zur Planung, Umsetzung, Kontrolle und Steuerung der Entwicklung der Unternehmensarchitektur umfasst (Niemann 2005). In der Literatur finden sich auch zu dem Begriff des Architekturmanagements unterschiedliche Perspektiven, die sich auf eine Veränderung des Fokus des Architekturmanagements im Laufe der Jahre zurückführen lassen. Diese Entwicklung wird in dem folgenden Unterabschnitt 2.2.5.1 aufgezeigt. Nach einer Darstellung der Ziele des Architekturmanagements in Unterabschnitt 2.2.5.2 wird in Unterabschnitt 2.2.5.3 die Standardisierung als eine etablierte Maßnahme des Architekturmanagements vorgestellt. Darüber hinaus wird diskutiert, warum dieser Ansatz nur eine Teilmenge der Ziele des Architekturmanagements berücksichtigt.

2.2.5.1 Die Entwicklung des Architekturmanagements

In den letzten 30 Jahren hat sich das Architekturmanagement von einem eher technischen – mit dem Fokus auf der Modellierung – zu einem die Geschäftsstrategie der Unternehmen berücksichtigenden Managementansatz weiterentwickelt (Ahlemann et al. 2012).

Die Wurzeln des Architekturmanagements sind dabei in Zachmans Framework für die Architektur von Informationssystemen zu finden (Zachman 1987). Dieses beinhaltet Empfehlungen dazu, welche architektonischen Aspekte (im Zachman Framework lauten diese Daten, Funktionen und Netzwerk) aus verschiedenen Perspektiven (u. a. Planer, Designer oder Eigentümer) bei der Ausgestaltung einer IT-Architektur herangezogen und modelliert werden sollten. In der Arbeit von Sowa & Zachman (1992) wurde das initiale Zachman Framework um organisationale Aspekte ergänzt (Menschen, Zeit und Motivation). Der auf der Modellierung der Architekturen liegende Fokus blieb jedoch erhalten. Im Laufe der 90er Jahre zeigte sich, dass ein rein modellbasierter, primär strukturbeschreibender Ansatz (vgl. hierzu auch Krcmar 1990; Scheer 1991) nicht ausreicht, um den Entwicklungen in der IT Herr zu werden (Ahlemann et al. 2012). Die immer weiter steigende IT-Durchdringung in den Unternehmen und die in diesem Kontext häufig durch die einzelnen Geschäftsbereiche dezentral getroffenen IT-Entscheidungen führten zur Entstehung sog. „Business Silos“. Diese zeichneten sich durch IT-Lösungen aus, die für den jeweiligen fachlichen oder geographischen Kontext zugeschnitten waren. Hierdurch entstehende positive Effekte (wie z. B. die Freiheitsgrade der Entwickler, sich stets aus den innovativsten und für den Anwendungsfall optimal passenden Technologien bedienen zu können) wurden in kurzer Zeit durch die gleichzeitig entstehenden Schwierigkeiten bei der Integration neuer Anwendungen in die bestehende Anwendungsarchitektur aufgewogen (Ross 2003). Aus dieser Problemstellung verschob sich die Rolle des EAM von der Modellierungssicht hin zu einer Management-Disziplin mit IT-Managementprozessen für die Planung, Umsetzung, Kontrolle und Steuerung der IT-Architektur. Anstelle einer Konzentration auf einzelne Anwendungen etablierte sich eine ganzheitliche Perspektive auf die Organisation der IT-Architektur des Unternehmens. Diese Entwicklung spiegelt sich auch in den Architekturrahmenwerken wieder, die zwar eine – häufig auf den Erkenntnissen von Zachman (1987) basierende – Form der Modellierung vorsehen, gleichzeitig aber auch Vorgaben für die Planung, Umsetzung, Kontrolle und Steuerung der IT-Architektur enthalten. Beispiele hierfür sind die auf dem Rahmenwerk von Zachman basierenden EAP (Spewak & Hill 1993), FEAF (The Chief Information Officers Council 1999) oder IAF (Integrated Architecture Framework; Wout et al. 2010). In der näheren Vergangenheit nach 2000 finden sich in der IS-Literatur zudem vermehrt Arbeiten, die sich Fragestellungen des Managements von Unternehmensarchitekturen und der Verbindung dieser mit der Unternehmensstrategie widmen. In diesen Arbeiten zeigt sich, dass ein erfolgreiches Architekturmanagement einer engen Abstimmung mit der Strategie des Unternehmens bedarf (z.B. Pulkkinen & Hirvonen 2005; Ross et al. 2006; Tamm et al. 2011). Dies wird nicht zuletzt auch aus der Integration des Literaturstranges des Business-IT Alignment deutlich (Bradley et al. 2012; Tamm et al. 2011).

2.2.5.2 Ziele des Architekturmanagements

Das in Abschnitt 2.2.2 eingeführte Verständnis einer Architektur im Kontext der Informationssysteme lässt vermuten, dass Architekturen relativ statische Konstrukte sind, die sich zwar weiterentwickeln, dies jedoch nach vorgegebenen Prinzipien. Da sich Organisationen stetig an ein sich verändertes (Geschäfts-)Umfeld anpassen, besteht auch an die Unternehmens- und IT-Architektur die Anforderung, sich diesen Änderungen anpassen zu können. Wie bereits angeführt, sind Planung, Umsetzung, Kontrolle und Steuerung der Entwicklung einer Architektur der Kern aktueller Architekturmanagement-Ansätze. Die durch das Architekturmanagement dabei verfolgten Ziele sind im Wesentlichen in der Unterstützung der Geschäftsziele durch die Bereitstellung und Aufrechterhaltung einer *(kosten-)effizienten und flexiblen* Unternehmens- bzw. IT-Architektur zu sehen (vgl. Ahlemann et al. 2012; Richardson et al. 1990; Ross et al. 2006; Schmidt & Buxmann 2011; Tamm et al. 2011). Für die weitere Arbeit werden diese Zielgrößen im Folgenden definiert.

2.2.5.2.1 IT-Flexibilität

Für den Begriff *Flexibilität* existiert keine allgemein akzeptierte Definition. Shewchuk & Moodie (1998) geben an, mehr als 70 verschiedene Definitionen in der Literatur identifiziert zu haben. In der Managementliteratur wird Flexibilität definiert als:

“[...] the degree to which an organization possesses a variety of actual and potential procedures, and the rapidity with which it can implement these procedures to increase the control capability of the management and improve the controllability of the organization over its environment.” (Byrd & Turner 2000, S. 170)

Aufbauend auf dieser Definition nehmen Tallon & Kraemer (2003) eine Differenzierung von IT-Flexibilität und strategischer Flexibilität vor:

“[...] IT flexibility reflects the ability to easily and quickly introduce new sources of IT support for the business strategy, while strategic flexibility reflects the ability to easily and quickly change key aspects of the business strategy in areas such as pricing, market reach, etc.” (Tallon & Kraemer 2003, S. 2)

Die vorliegende Arbeit orientiert sich an dem Begriffsverständnis von „IT-Flexibilität“ nach Tallon & Kraemer (2003) und definiert diese wie folgt:

IT-Flexibilität ist die Eigenschaft einer IT-Architektur – mit angemessenem Aufwand – an neue (Geschäfts-)Anforderungen angepasst werden zu können.

2.2.5.2.2 IT-Effizienz

Den Begriff der *Effizienz* beschreibt Drucker (1986) aus einer organisationalen Perspektive als:

“[...] doing better what is already being done.” (Drucker 1986, S. 36)

Eine vergleichbare Tendenz in Richtung der Optimierung, jedoch mit explizitem Fokus auf dem Ressourceneinsatz, bzw. den Kosten zeigt die Definition von Butler (1999):

„Efficiency refers to the minimization of resources used to discover and implement solutions.“ (Butler 1999, S. 221)

In Anlehnung an die angeführte Ressourcen- bzw. Kostenperspektive wird die folgende Definition des Begriffs „IT-Effizienz“ verwendet:

IT-Effizienz ist die Eigenschaft einer IT-Architektur, bestehende Geschäftsanforderungen mit möglichst geringem Ressourceneinsatz erfüllen zu können.

2.2.5.3 Standardisierung – eine Maßnahme des Architekturmanagements

Eine häufig in der Forschung diskutierte und in der Praxis verwendete Maßnahme für das Architekturmanagement ist die der Standardisierung. In der vorliegenden Arbeit wird der Begriff der Standardisierung im Sinne der Definition der „Standardisierung in betrieblichen Informationssystemen“ nach Buxmann (1996) verwendet:

„Unter Standardisierung in betrieblichen Informationssystemen wird eine Entscheidung über den Einsatz von Standards zur Vereinheitlichung von Systemelementen verstanden [...]“ (Buxmann 1996, S. 10)

Ein Standard ist dabei – der Definition von De Vries (2006) folgend – zu verstehen als:

“[...] approved specification of a limited set of solutions to actual or potential matching problems, prepared for the benefits of the party or parties involved, balancing their needs, and intended and expected to be used repeatedly or continuously, during a certain period, by a substantial number of the parties for whom they are meant.“ (De Vries 2006, S. 4)

In der Praxis finden häufig Standardkataloge (sogenannte „Books of Standards“) Anwendung, in denen Produkte und Technologien dokumentiert sind, die im Zuge einer Neuentwicklung oder Änderung für den Einsatz im Unternehmen freigegeben sind (vgl. Schmidt 2009; The Open Group 2011). In Tabelle 4 ist eine Auswahl wissenschaftlicher Veröffentlichungen im Feld der Unternehmensarchitekturen angeführt, in denen (IT-) Standardisierung eine wesentliche Rolle spielt.

Die bestehende Forschung zur IT-Standardisierung verwendet zur Lösung von Standardisierungsproblemen meist mathematische (Optimierungs-)Modelle, in deren Zielfunktionen die Reduktion von Kosten und somit die Erhöhung der IT-Effizienz eine bedeutende Rolle spielt (Buxmann et al. 1999; Li & Chen 2012; Weitzel et al. 2006; Widjaja 2010). Eng hiermit verbunden ist der Gedanke, dass eine Reduktion der Heterogenität der IT-Architektur i. d. R. wünschenswert ist. Dies ist damit zu begründen, dass ein Abweichen von innerhalb eines Unternehmens vorgegebenen IT-Standards – wie beispielsweise den in den o. a. Standardkatalogen dokumentierten Technologien – mögliche Synergieeffekte nachteilig beeinflusst (Boh & Yellin 2006; Guillemette & Paré 2012; Richardson et al. 1990). Hieraus wird zudem deutlich, dass sich IT-Standardisierung stark an dem in der IS-Forschung etablierten ressourcenbasier-

ten Ansatz orientiert. Sie berücksichtigt hingegen nicht die sich stetig ändernden Anforderungen durch die Geschäftsseite (Allen & Varga 2006).

IT-Standardisierung hat demzufolge einen starken Fokus auf der Erhöhung der IT-Effizienz durch einen kostenoptimierten Ressourceneinsatz. Eng mit der IT-Standardisierung verbunden ist eine Reduktion der Heterogenität der in einer IT-Landschaft vorherrschenden Technologien.

Tabelle 4: IS-Forschungsbeiträge im EA-Kontext mit Standardisierungs-Linse

Quelle	Forschungsansatz	Relevante Ergebnisse für die vorliegende Arbeit
Richardson et al. (1990)	Fallstudie und Erarbeitung von Designprinzipien zur Ausgestaltung der IT-Architektur	▪ Einführung des Gedankens einer prinzipienbasierten Unternehmens- bzw. IT-Architektur ▪ Eine homogene [standardisierte] Architektur verbessert Informationsflüsse sowie die Support-Kosten des Systems
Ross et al. (2006)	Fallstudien bei 69 Unternehmen (aus den USA und Europa) zu Themen des Unternehmensarchitekturmanagements	▪ Etablierung einer neuen, strategischen Perspektive auf das Management von Unternehmensarchitekturen ▪ Hervorhebung von 1) IT-Standardisierungs- und Integrationsentscheidungen bei der Auflösung des Wildwuchses historisch gewachsener IT-Architekturen und 2) der Ausgestaltung wiederverwendbarer Plattformen
Boh & Yellin (2006)	Empirische Untersuchung 1) des Einflusses von Governance- Mechanismen auf EA-Standards sowie 2) der Wirkung des Einsatzes von EA-Standards in den Architekturebenen	▪ Unterstützende Wirkung der Nutzung von IT-Standards bei dem Management der IT-Infrastruktur-, Anwendungs- sowie Datenressourcen ▪ Herstellung der Verbindung zwischen IT-Standardisierung und der Reduktion von Heterogenität und Redundanz sowie der zunehmenden Integration der IT-Architektur
Schmidt & Buxmann (2011)	Konzeptentwicklung sowie empirische Untersuchung der Wirkung eines Managements von Unternehmensarchitekturen	▪ Vorschlag einer Konzeptualisierung des Managements von Unternehmensarchitekturen ▪ Nachweis der Wirkung eines Managements von Unternehmensarchitekturen auf IT-Effizienz und IT-Flexibilität ▪ Herstellung der Verbindung des Managements von Unternehmensarchitekturen mit der Reduktion von IT-Komplexität und einer intensiveren Nutzung von Standardisierung¹⁷
Li & Chen (2012)	Konzeptentwicklung und mathematische Formulierung von IT-Standardisierungsproblemen	▪ Betonung von IT-Standardisierung als wesentliche Aufgabe des IT-Managements ▪ IT-Standardisierung führt zu Kosteneinsparungen im administrativen Bereich der IT und steigert die Kompatibilität zwischen den eingesetzten Produkten ▪ IT-Standardisierung geht zu Lasten des Wunsches der Nutzer nach Vielseitigkeit

Die zweite der in Abschnitt 2.2.5.2 eingeführten Zielgrößen des Architekturmanagements, die IT-Flexibilität, steht bei der IT-Standardisierung jedoch nicht im Vordergrund. Dies wird

¹⁷ Standardisierung findet sich in drei der sieben von Schmidt & Buxmann (2011) eingeführten Indikatoren des Konzeptes EAM wieder.

auch durch die gezielte Reduktion von Heterogenität deutlich: Während Heterogenität von Technologien für die Erreichung eines Kostenoptimums – z. B. durch eine Erhöhung der Informationskosten aufgrund mangelnder Kompatibilität der Schnittstellen – tendenziell eher schädlich ist, kann diese aus einer IT-Flexibilitäts-Perspektive durchaus erwünscht sein. So kann eine größere Heterogenität die Einführung einer neuen Lösung auch begünstigen, da im Zuge der Umsetzung auf eine größere Auswahl verfügbarer Technologien zurückgegriffen werden kann. Heterogenität kann also nicht per se als etwas schädliches angesehen werden. Vielmehr ist ein gewisser Grad an Heterogenität erforderlich, um sowohl gegebene als auch neu hinzukommende Anforderungen umsetzen zu können. Eine Aufgabe des Architekturmanagements besteht demnach darin, einen angemessenen Grad an Heterogenität zu identifizieren.

2.2.6 Die Rolle des Architekten

Die Rolle in der Organisation, die sich in erster Linie dem Management der Unternehmens- bzw. IT-Architektur widmet, ist die des Architekten¹⁸. Der IEEE Standard 1471–2000 definiert den Architekten (engl. „architect“) als:

„The person, team, or organization responsible for systems architecture.“ (IEEE Architecture Working Group 2000, S. 3)

Ahlemann et al. (2012) spezifizieren das Aufgabenspektrum des Architekten durch die Verwendung der Analogie zur Städteplanung¹⁹. Ihnen zufolge muss ein Architekt u. a. zukünftige Anforderungen antizipieren, Pläne erstellen, verschiedene Interessengruppen zusammenbringen und eine ganzheitliche – verschiedene Zielgrößen berücksichtigende – Perspektive einnehmen (Ahlemann et al. 2012, S. 8).

Guillemette & Paré (2012) führen das Management der Komplexität der IT-Architektur als eine der fundamentalen Aufgaben der IT-Abteilungen in Unternehmen an. Für das hierfür verantwortliche Profil verwenden Guillemette & Paré (2012) den Begriff „Architecture Builder“. Aufgabe des „Architecture Builders“ ist es dabei:

“[...] to build and manage an IT infrastructure that supports business processes and reduces architectural complexity.” (Guillemette & Paré 2012, S. 532).

Die im Zusammenhang mit diesen Aufgaben verbundenen Aktivitäten sollen neben der *Reduktion der Komplexität der IT-Architektur* (z. B. durch Standardisierung) auf die *Entwicklung einer integrierten und flexiblen IT-Architektur* abzielen (vgl. Unterabschnitt 2.2.5.3).

¹⁸ In der Praxis finden sich unterschiedliche Rollenbezeichnungen wie z. B. „Enterprise Architect“, „IT Enterprise Architect“ oder „IT Architect“. Da Architektur-Initiativen häufig durch die IT-Bereiche getrieben werden, befassen sich die Mitarbeiter in den angeführten Rollen grundsätzlich mit Themen im IT-nahen Umfeld der IT-Architektur und den aus dieser Perspektive heraus bestehenden Schnittstellen zur Geschäftsprozessebene.

¹⁹ Die Metapher der Städteplanung wird in einer Reihe von Arbeiten verwendet und diskutiert (Namba & Iijima 2004; Schmidt & Buxmann 2011). Eine kritische Betrachtung der Analogie zwischen der Städteplanung und dem Enterprise Architecture Management führt Ross (2003) an. Ihr zufolge wird hierdurch lediglich eine eher technologische Perspektive auf die Beziehung zwischen der IT und den Geschäftsprozessen abgebildet. Das strategische Potenzial einer Unternehmensarchitektur bleibt Ross zufolge außen vor.

2.3 Theoretischer Bezugsrahmen der Arbeit

In diesem Abschnitt werden die in 2.1 und 2.2 eingeführten Konzepte zu einem theoretischen Bezugsrahmen für die vorliegende Arbeit zusammengeführt (vgl. Abbildung 7).

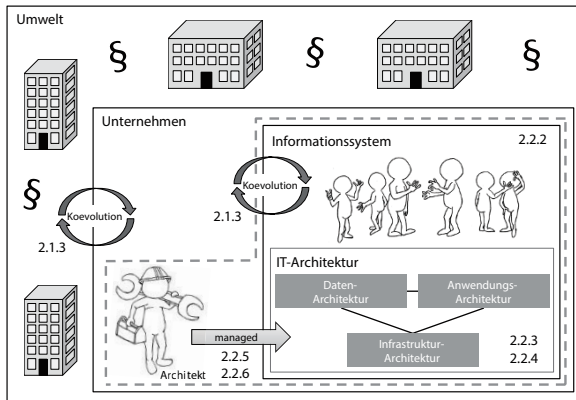


Abbildung 7: Theoretischer Bezugsrahmen der Arbeit

Wie in Abschnitt 2.2.2 dargestellt, wird im Kontext von Informationssystemen ein an die Systemtheorie (vgl. Abschnitt 2.1.1) angelehnter Architekturbegriff verwendet. Eine Architektur wird definiert als die grundlegende Struktur eines künstlichen, durch Menschen erstellten, gepflegten und weiterentwickelten Systems, das aus Komponenten und den Beziehungen zwischen den Komponenten besteht. Informationssysteme werden in der wissenschaftlichen Literatur häufig als CAS betrachtet. Somit müssten Informationssysteme den in Tabelle 1 angeführten System-Charakteristiken, wie u. a. Selbstorganisation, Emergenz und Koevolution folgen. Wie in Abschnitt 2.1.3 angeführt, zeigt die IT-Architektur (vgl. Abschnitte 2.2.3 und 2.2.4) jedoch keine agentenhaften Charakteristiken, wie sie die Theorie der CAS unterstellt. Speziell die Charakteristik der Koevolution ist hier nicht zutreffend. Um eine solche Koevolution der IT-Architektur sicherzustellen, hat ein Agent im Umfeld des Informationssystems – der IT-Architekt (Abschnitt 2.2.6) – die Aufgabe, die IT-Architektur in der Form aufzubauen und zu managen, dass sie den sich ändernden Anforderungen seiner Umwelt genügt und dabei seine Anpassbarkeit (IT-Flexibilität) nicht verliert. Abschnitt 2.2.5 zeigt, dass bestehende Strategien (speziell die der IT-Standardisierung) primär auf die Erreichung von IT-Effizienz abzielen, das Ziel einer IT-Flexibilität jedoch außer Acht lassen.

Die übergeordnete Fragestellung der Arbeit ist demnach, wie die IT-Architektur hinsichtlich einer sich im ständigen Wandel befindlichen Umwelt gemanagt werden kann. Hierzu soll im Folgenden mit der IT-Komplexität – bzw. genauer der strukturellen Komplexität von IT-Architekturen – eine neue, die bestehenden Perspektiven komplementierende Perspektive untersucht werden. Der Fokus der vorliegenden Arbeit liegt dabei auf dem in Abbildung 7 durch die gestrichelte Linie eingerahmten Teilbereich. Im folgenden Kapitel 3 wird der Begriff der Komplexität in der IS-Literatur untersucht und auf Grundlage einer strukturierten Literaturrecherche die weitere Forschungsagenda der Arbeit abgeleitet.

Komplexität von IT-Architekturen
Konzeptualisierung, Quantifizierung, Planung und
Kontrolle

Schütz, A.

2017, XXII, 240 S. 65 Abb., Softcover

ISBN: 978-3-658-18225-0