Clustering von Anwendungslandschaften mit Self-Organizing Maps (SOMs)

Daniel Fürstenau

DFG-Graduiertenkolleg „Pfade organisatorischer Prozesse“
Freie Universität Berlin
Garystr. 21
14195 Berlin
daniel.fuerstenau@fu-berlin.de

Abstract: Es besteht ein hohes Interesse an Techniken zur Analyse von Anwendungslandschaften. Im Sinne eines Portfoliomanagements sind für diese Aufgabe häufig mehrere Dimensionen relevant. Werden Kennzahlen verwendet, entstehen multidimensionale Datensätze, die sich bei einer sehr großen Anzahl an Elementen mittels konventioneller Techniken nur noch schwer erschließen lassen. Die *Self-Organizing Map (SOM)* ist ein vielversprechender Ansatz aus dem Bereich der Clustering-Methoden mittels künstlicher neuronaler Netzwerke. In dieser Arbeit wird die SOM als Steuerungsinstrument für das IT-Management verwendet. Anhand von Kennzahlen wird ein Clustering ähnlicher Anwendungen in einem Portfolio durchgeführt. Es sollen so Zeit und Aufwand für den Analysten eingespart werden. Die Anwendungen im Portfolio lassen sich anhand ihrer Cluster-Zugehörigkeit gezielt vergleichen. Es wird gezeigt, welche Erfahrungen gewonnen wurden und welche Implikationen sich für Forschung und Praxis ergeben.

1 Einleitung

Für Portfoliobetrachtungen des IT-Managements und für die strategische IT-Planung bilden die Anwendungen eines Unternehmens eine wichtige Analyseeinheit (vgl. u.a. [Kü06], [Ma06], [EV08], [De09], [Du10]). Die Anzahl der Anwendungen in der Anwendungslandschaft von großen und mittleren Organisationen beträgt dabei häufig mehrere Hunderte bis Tausende (vgl. [Ke06], [Ai+09], [Du09]). Im Sinne einer zielgerichteten IT-Steuerung soll dieses Portfolio unter verschiedenen Aspekten, z.B. Kosten, Wertbeitrag und Qualität, vermessen, analysiert und schließlich optimiert werden [Kü06]. Ausgehend von den methodischen Herausforderungen für den Analysten, der mit Daten über ein solches Portfolio operiert, gilt meine Aufmerksamkeit in diesem Beitrag den folgenden Fragen: *Auf welche Art und Weise können in diesen großen Datenmengen automatisiert Gruppen von Anwendungen mit ähnlichen Merkmalen gefunden werden? Welchen Beitrag kann die Self-Organizing Map, eine spezielle Methode aus dem Bereich der künstlichen Intelligenz, für diese Aufgabe leisten?* Die Frage ist eng eingebunden in den Prozess des *Anwendungsportfoliomanagements* [Kü06]. Er unterliegt dem *Regelkreis* aus Planung, Umsetzung, Analyse und Korrektur [Kü06].

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

Traditionell (vgl. [Ke06]) liegen die Informationen über die Anwendungen oft nur verstreut oder in Form von Anwendungshandbüchern vor. Zur Bewertung des Zustands der Anwendungen überprüfen die Verantwortlichen in vielen Fällen einzelne Anwendungen und versuchen manuell für jede der Anwendungen eine Kategorisierung vorzunehmen. Hierunter soll verstanden werden, dass einzelne Anwendungen aufgrund der Einschätzung eines Experten als *gut*, *mittel* oder *schlecht* bewertet werden. Aufgrund des Aufwands zur Durchführung des Verfahrens erfolgt mitunter eine Beschränkung auf eine Teilmenge der Anwendungen. Die Bewertung wird teilweise durch dezentrale Experten vorgenommen und später zusammengeführt. Bei einer geringen Anzahl an Merkmalen und überschaubaren Datenmengen lassen sich die Informationen noch verhältnismäßig einfach erheben und mit Standardanwendungen, wie MS Excel, analysieren und visualisieren [Kü06]. Bei einer größeren Anzahl an Merkmalen und größeren Datenmengen stößt der Nutzer allerdings an die Grenzen der Interpretierbarkeit und Standardanwendungen, aber auch spezialisierte Unternehmensarchitekturmanagement (UAM)-Werkzeuge (vgl. [Ma08a]), an die Grenzen der graphischen Darstellbarkeit. Subjektive Einflüsse spielen bei der Bewertung eine (zu) große Rolle und es erfolgt eine Konzentration auf die bekannten Kandidaten, wobei unbekannte Zusammenhänge nicht ausreichend gewürdigt werden. Gesamtaussagen über den Zustand des Portfolios lassen sich nur schwer treffen.

In vielen Fällen kann eine Bewertung mittels *Kennzahlen* den IT-Manager bei seiner Steuerungsaufgabe unterstützen [Kü06]. Um über die Gesamtheit der relevanten Merkmale eines Steuerungsobjekts zu informieren, werden in vielen Fällen auch *Kennzahlensysteme* eingesetzt [Kü06]. Durch die Verwendung solcher Kennzahlensysteme entstehen multidimensionale Datensätze, die mehrere relevante Dimensionen enthalten. *Statistische Methoden* sollen die Aufgabe der Analyse solcher multidimensionaler Datensätze unterstützen [Le10]. Zudem bieten diese Interpretationshilfen und Visualisierungen für die Anwender [Le10]. Eine wesentliche Herausforderung in solchen Datensätzen besteht im Auffinden von ähnlichen Objekten und deren Zuordnung zu einer gemeinsamen Gruppe (= Clustering) [Le10]. Objekte *innerhalb* einer Gruppe sollen so *homogen* wie möglich sein. Im Gegensatz zu *anderen* Gruppen soll das Cluster so *heterogen* bzw. *isoliert* wie möglich sein. Entsprechende statistische Methoden werden als Clustering-Methoden bezeichnet [Le10].

Im Folgenden argumentiere ich, dass mit der *Self-Organizing-Map (SOM)* [Ko95] das Clustering von Anwendungen in einem Portfolio bei großen Datenmengen verbessert werden kann. Das Ziel eines solchen Clusterings liegt in der Unterstützung für das IT-Management bei seiner Steuerungsaufgabe. Innerhalb des Portfolios lassen sich einfacher Gruppen mit gleichartigen Objekten erkennen. Die Objekte *dieser* Gruppe lassen sich von *anderen* Gruppen abgrenzen. Systematische Unterschiede zwischen den einzelnen Gruppen werden analysierbar. So lässt sich ein Benchmarking [Kü06] innerhalb des Portfolios realisieren. Hierunter verstehe ich ein Vorgehen, bei dem eine Anwendung auf Basis der Zuordnung zu einer Gruppe mit anderen *verglichen* wird. Die Resultate des Clusterings erlauben relative Schlussfolgerungen, wie im folgendem Beispiel: Anwendung A gehört zur Gruppe der Anwendungen mit *schlechten* Werten über alle Dimensionen während dies für Anwendung B nicht gilt. Unter der Voraussetzung funktionaler Äquivalenz sollte Anwendung B zukünftig bevorzugt werden.

Im Gegensatz zu anderen Clustering-Ansätzen für das IT-Management (vgl. [Ho+08], [AW08]) werden *Kennzahlen* für das Clustering verwendet. In Abgrenzung zu anderen kennzahlenbasierten Ansätzen werden allerdings keine Verknüpfungen zwischen Kennzahlen [Kü06] betrachtet. Mehrdimensionale Merkmalsräume bilden hier lediglich die Basis für das Clustering. Die erzielten Resultate sind zunächst dem oben beschriebenen, traditionellen Vorgehen überlegen, da Fehlerquellen verringert werden können, indem die Daten durch automatisierte Prinzipien voranalysiert werden [Le10]. Bisher unbekannte Zusammenhänge in den Daten die der Intuition verborgen bleiben, können aufgedeckt werden [Le10]. Für den Analysten vereinfacht sich die Analyse, da die Abhängigkeit von Fachexperten verringert wird. Fachexperten wiederrum werden durch die Voranalyse entlastet und werden eher als prüfende Instanz eingebunden. Die entstehenden Darstellungen lassen sich als Instrument der Kommunikation mit Fachexperten und dem Management verwenden. Zu kennzahlenbasierten Methoden steht das Clustering in einem komplementären Verhältnis. Es kann als Ergänzung für den Analysten bei seiner Aufgabe gesehen werden.

Self-Organizing Maps, eine Form neuronaler Netzwerke, sind eine unbeaufsichtigte Lernmethode [Ek+02]. Im Unterschied zu beaufsichtigten Lernmethoden wird der Vorschlag für das Clustering ausschließlich anhand der vorliegenden *Daten* gegeben. Daten müssen bei der SOM, z.B. im Gegensatz zur Diskriminantenanalyse, nicht vorklassifiziert werden [Ab07]. Die Auswahl der SOM als Clustering-Ansatz erfolgte anhand folgender Überlegungen: *(1)* Die SOM erfährt in der wissenschaftlichen Anwendung eine hohe Popularität. Das Adaptive Informatics Research Center in Helsinki, eines der führenden Institute auf diesem Gebiet, verzeichnet aktuell 7718 Arbeiten mit der SOM-Technik [AIR11]. Zu den Anwendungsgebieten zählen z.B. Mustererkennung in der Bioinformatik und Robotik, Data Mining und Texterkennung oder die Diagnose bei Krankheiten. Im organisatorischen Bereich gehören zu den Anwendungen u.a. das Assessment von Kreditrisiken und die Vorhersage von Finanzkrisen [JS11] und ein Benchmarking auf Basis finanzieller Kennzahlen [Ek+02]. Insbesondere das letztgenannte Beispiel geht von einer sehr ähnlichen Problematik aus *(2)* Bei klassischen Clustering-Aufgaben zeichnen sich die Ergebnisse der SOM durch hohe Genauigkeit aus (vgl. [Ab07], [Ba+05], [Tr08], [KK01]). Ein anschauliches Beispiel für die Anwendbarkeit für nichttriviale Probleme ist der *Fisher Iris Datensatz.* Hierbei erfolgt auf Basis eines multivariaten Datensatzes eine Zuordnung zu drei Clustern [Ve+99]. *(3)* Die SOM lässt sich mit anderen Clustering-Methoden kombinieren (vgl. z.B. [Ba+05]). Auch für die Visualisierung lassen sich alternative Techniken, z.B. die Hauptkomponentenanalyse, anwenden (vgl. [Ve+00]).

Die nachfolgenden Teile sind wie folgt aufgebaut: *Kapitel 2* erläutert die Methodik und stellt den Bezug zum UAM her. *Kapitel 3* zeigt an einem Beispiel, wie die vorgestellte Methode angewendet werden kann. *Kapitel 4* befasst sich basierend auf den Erfahrungen mit praktischen Implikationen. *Kapitel 5* schließt mit einem Fazit.

2 Methodik

2.1 Theoretische Grundlagen

Bei der *SOM*, auch als *Kohonen Map* [Ko95] bezeichnet, wird ein multidimensionaler Datensatz auf eine zweidimensionale Karte (engl. Map) reduziert [Ek+02]. Die Karte besteht aus sog. Neuronen [Ve+00]. Jedes *Neuron* wird repräsentiert durch einen *d*-dimensionalen Gewichtsvektor $m = [m_1,..,m_d]$. Die Gewichtungsvektoren der SOM sind typischerweise multidimensional. Zur Abbildung der Gewichtungsvektoren auf die Map fungiert die SOM auch als Vektorprojektionsalgorithmus, der die Dimensionalität reduziert [Ka97].

Nachbarschaftsfunktionen *verbinden* Neuronen mit seinen Nachbarn. Die Nachbarschaft zwischen Neuronen bestimmt die *Topologie* des Netzwerkes und hiermit die Form des *Gitters* [Ve+00]. Bei einem hexagonalen Gitter ist jedes Neuron mit sechs Nachbarn verbunden [Ve+00]. Abbildung 1 zeigt eine rechteckige SOM-Map mit einem hexagonalen Gitter. In diesem ist der Nachbarschaftsradius eines Neurons mit seinen direkten Nachbarn eingezeichnet.

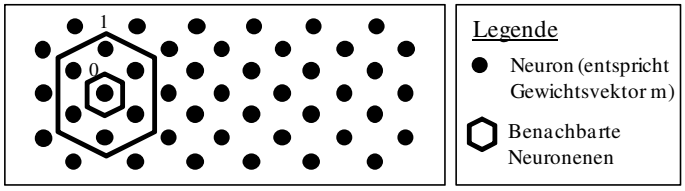

Abbildung 1: SOM-Map mit hexagonalem Gitter. Ein Neuron mit Nachbarn wurde markiert. Abbildung in Anlehnung an Vesanto et al. [Ve+00, S. 8]

Die *Funktionsweise* des SOM-Algorithmus lässt sich wie folgt beschreiben (vgl. [Ve+00]): (1) Es wird zufällig ein Gewichtsvektor x aus der Menge der Gewichtsvektoren ausgewählt. (2) Für diesen Gewichtsvektor x wird die Distanz zwischen allen anderen Gewichtsvektoren $m = [m_1,\ldots,m_d]$ und x berechnet. Das *Neuron* m_c mit der minimalen Distanz gewinnt. Dieses wird auch als *Best Matching Unit (BMU)* bezeichnet. (3) Nach dem Auffinden der BMU werden die Gewichtsvektoren der SOM angepasst, so dass die BMU enger an den Gewichtsvektor x heranrückt. Abbildung 2 zeigt beispielhaft diese Anpassung. Die Veränderung des Gewichtsvektors der BMU in Richtung x erfolgt gemäß einer Lernrate α, Gewichtsvektoren die über die Nachbarschaftsfunktion h_{ci} mit dem BMU verbunden sind, werden auch, aber weniger stark, in Richtung von x geändert. Im Ergebnis werden die Neuronen so in eine Reihenfolge gebracht und benachbarte Neuronen haben ähnliche Gewichtsvektoren [Ve+00]. (4) Die Schritte eins bis drei werden für eine vorgegebene Anzahl an *Iterationen* wiederholt.

ISBN 978-3-88579-286-4

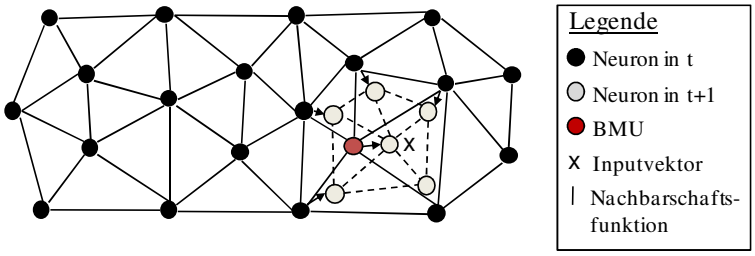

Abbildung 2: Bewegung der Best Matching Unit (BMU) in Richtung des Gewichtsvektors x. Durchgezogene und gestrichelte Linien entsprechen Situation vor und nach dem Training. Abbildung in Anlehnung an Vesanto et al. [Ve+00, S. 9]

Nach dem Training sind als *Visualisierungs- und Interpretationshilfe* weitere Techniken notwendig. Zur Visualisierung in der SOM wird die sog. *U-Matrix* verwendet [Ek+02]. Die U-Matrix fügt auf einer zweidimensionalen SOM als dritte Dimension die durchschnittliche Entfernung des Gewichtsvektors eines Neurons zu den Gewichtsvektoren seiner direkten Nachbarn hinzu. Übertragen auf eine Farbskala zeichnen sich die Clustergrenzen auf der entstehenden Karte farblich ab und lassen sich so erkennen [Ko+96]. „Wasserscheiden“ unterteilen die U-Matrix in Cluster [US90]. Zum vertieften Studium der Daten werden außerdem die einzelnen Variablen auf sog. *Feature Plan Maps* separat visualisiert. Ergänzend lassen sich weitere Visualisierungen, z.B. graphische Funktionen oder Scatterplots, verwenden [Ve+00].

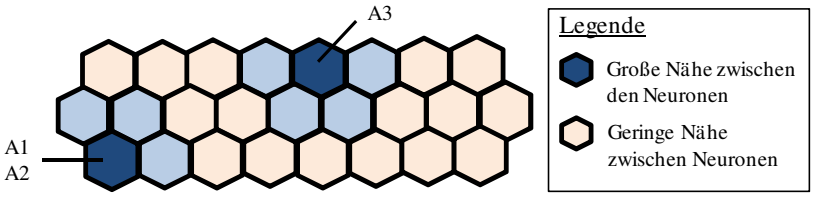

Abbildung 3: U-Matrix

Die Ergebnisse werden dem Analysten präsentiert. Zur Beurteilung steht u.a. das Maß der *kleinsten quadratischen Abweichung* zur Verfügung. Ist der Analyst mit den Ergebnissen nicht zufrieden, wird das Training mit geänderten Parametereinstellungen, insbesondere Lernrate und Größe der Map, wiederholt. Formel 2 wird auch als *Kohonen Lernregel* bezeichnet [US90]. Die Lernrate α bestimmt dabei *wie stark* ein Gewichtsvektor verändert wird, d. h. wie stark das Neuron lernt [Ve+00].

Gegenüber anderen Clustering-Methoden werden in der Literatur Vor- und Nachteile der SOM diskutiert (vgl. [SS04], [Ba+05], [Ab07], [Le10]). In einem Vergleich mit vier anderen Clustering-Methoden (K-Means, Hierarchical Clustering, Expectation Maximation Clustering) folgert Abbas [Ab07]: „*SOM shows more accuracy in classifying most of the objects into their suitable clusters than other algorithms*“ und “*all the algorithms have the some ambiguity in some noisy data when clustered”*.

Bacao et al. [Ba+05] untersuchen in einer Studie die SOM als Alternative zu K-means Clustering für vier traditionelle Datenanalyse-Aufgaben. Sie folgern, SOM „*shows a tendency to outperform K-means*“ [Ba05]. Weitere Autoren schlagen eine Erweiterung oder Kombination der SOM mit anderen Clustering-Ansätzen vor, um bestimmte *Nachteile* der SOM auszugleichen. In diesem Zusammenhang stehen vor allem die ex-post Interpretierbarkeit der SOM und die U-Matrix im Mittelpunkt. Trosset [Tr08] schlagen eine Integration der SOM mit dem *Multidimensional Scaling* vor, um die *Interpretierbarkeit* der SOM zu verbessern. Shahakapurkar et al. [SS04] entwickeln die SOM weiter, um Problemen der *Reabilität* und *Interpretierbarkeit* zu begegnen. Auf eine eigene Bewertung wird im Rahmen der Implikationen näher eingegangen.

2.2 Forschungsmethodik

In dem Beitrag wird eine *Design Science* Perspektive eingenommen [He+04]. Auf Basis eines organisationalen Problems, hier das Clustering von Anwendungen in einem Portfolio (vgl. Abschnitt 1), schlage ich eine alternative Lösung vor. Die Lösung umfasst die Anwendung einer theoretisch entwickelten Methode und deren Einbettung in ein neues Vorgehen in einem bisher nicht erschlossenen Anwendungsfeld. Diese Lösung wende ich auf einen Datensatz an und diskutiere anhand eines fiktiven Falls Implikationen. Die vorgeschlagene Evaluierung soll weitere Forschung befördern.

2.3 Entwicklung der vorgeschlagenen Methode

Wie in Abschnitt 1 dargelegt, steht der Analyst bei der Analyse des Anwendungsportfolios eines Unternehmens vor einer herausfordernden Aufgabe. Er muss große Datenmengen in mehreren Dimensionen auswerten, um Vergleiche anzustellen. Dabei fehlt es ihm mitunter an Hintergrundwissen, da dieses nicht in zugreifbarer Form vorliegt. Um die Probleme des traditionellen Vorgehens zu umgehen, schlage ich ein alternatives Vorgehen vor. Grundlage dabei ist das Management des Anwendungsportfolios in ein ganzheitliches UAM zu integrieren. Unter UAM wird hier in Anlehnung an Aier et al. [ARW08] ein *modellbasierter* Ansatz verstanden, der die wesentlichen *Objekte* eines Unternehmens und deren *Beziehungen* erfasst und analysiert [ARW08].

Die Erfassung der Objekte erfolgt in einer integrierten Datenbasis. Erst auf einer solchen Datenbasis kann die Anwendung von Clustering-Methoden gelingen. Die Betonung der *Beziehungen zwischen Elementen* spielt eine herausgehobene Rolle. Sie ermöglichen es, *Kennzahlen* über die Objekte mittels des Beziehungswissens zu berechnen. Im Folgenden verwende ich als Grundlage ein Ausschnitt aus einem UAM-Metamodell [EFK11], um dies zu demonstrieren (vgl. Abbildung 4). Dieses besteht aus *Anwendungen*, *Funktionen* und *Schnittstellen* sowie die Beziehungen der Objekttypen untereinander. Ein Weg der Anreicherung des Metamodells um Kennzahlen bezieht sich auf die *direkte Zuordnung* von Kennzahlen zu einem Objekttyp (z.B. *Wartungskosten* zu *Anwendungen*). Eine weiterer Weg besteht in der Berechnung von (relationalen) Kennzahlen anhand der Anzahl und Art der Beziehungen zu den benachbarten Objekttypen (z.B. Anzahl der bereitgestellten Funktionen einer Anwendung). Basierend auf dieser Überlegung werden nachfolgend drei Kennzahlen entwickelt, um dies zu erläutern.

ISBN 978-3-88579-286-4

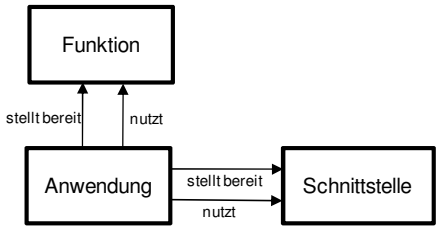

Abbildung 4: Ausschnitt aus UAM-Metamodell in Anlehnung an Eckert et al. 2011

Es werden Kennzahlen für die Messung der Konstrukte *Komplexität* und *Kosten* der Anwendungen definiert. Die Kennzahlen sind *exemplarisch* und von *intendiert geringer Komplexität*. Eine Einschränkung der Anwendbarkeit der SOM-Methode ist hiermit nicht verbunden. Der *exemplarische Charakter* ist nur insofern von Relevanz für die Anwendung der SOM, dass über die gewählten Merkmalsdimensionen Aussagen möglich sein müssen. Eine *sinnvolle* Vergleichbarkeit und Interpretierbarkeit intern homogener, extern aber heterogener Cluster wäre sonst nicht gegeben. Die *intendiert geringe Komplexität* ist für die SOM nicht relevant. Eine relevante Anforderung der Methode sind aber intervallskalierte Merkmalsausprägungen über alle Merkmale.

1) Funktionale Abhängigkeit: In der Literatur werden verschiedene Vorschläge zur Messung des Funktionsumfangs von Anwendungen diskutiert (vgl. [Ma05b], [Kü06]), z.B. definiert Masak [Ma05b] eine Funktionspunkt-Metrik, um den Umfang einer Anwendung unter Berücksichtigung der Komplexität der Funktionen zu messen. Basierend auf diesen Überlegungen wird in *Formel 1* die Kennzahl der funktionalen Abhängigkeit definiert. Anwendungen stellen Funktionen bereit und nutzen Funktionen anderer Anwendungen (vgl. Abbildung 4).

$$FC_{Anwendung} = f(Anzahl\ bereitgestellter\ Funktionen, Anzahl\ genutzter\ Funktionen) \quad (1)$$

Es wird vermutet, dass mit steigender Zahl bereitgestellter und genutzter Funktionen die Komplexität einer Anwendung steigt. Häufig findet sich eine Zahl von ca. einer bis 15 bereitgestellten Funktionen pro Anwendung, wobei die Verteilung eher linksschief ist, d.h. viele Anwendungen mit einer oder wenigen Funktionen stehen wenigen Anwendungen mit vielen Funktionen gegenüber.

2) Interface Komplexität (IC): Für die Integration von Anwendungen in vernetzten Anwendungslandschaften ist der Begriff der Schnittstelle etabliert [En+08]. Das Metamodell in Abbildung 4 trägt diesem Sachverhalt Rechnung. Basierend hierauf wird in Formel 2 die Kennzahl der Interface Komplexität definiert.

$$IC_{Anwendung} = f(Anzahl\ bereitgestellter\ Schnittstellen, Anzahl\ genutzter\ Schnittstellen) \quad (2)$$

Mit steigender Zahl bereitgestellter und genutzter Schnittstellen steigen die Abhängigkeit einer Anwendung von anderen und damit die Komplexität. Häufig sind 60 modellierte Schnittstellen für eine Anwendung nicht ungewöhnlich. Der Großteil der Anwendungen hat eher aber zwischen einer und zehn Schnittstellen.

ISBN 978-3-88579-286-4

3) Wartungskosten: In der Literatur werden verschiedene Dimensionen des Kostenbegriffs für Anwendungen diskutiert (vgl. [Kü06], [Kü10], [Ma05b]). Hier verwende ich die Wartungskosten aufgrund ihrer Bedeutung [Ma05b] stellvertretend für eine differenzierte Betrachtung dieses komplexen Konstrukts.

Vor der Analyse mittels der SOM sind die entsprechenden Kennzahlten festzulegen und zu erheben. Basierend auf den beschriebenen Grundlagen schlage ich das nachfolgend beschriebene Vorgehen vor. Die Analyse ist ggf. iterativ zu einem späteren Zeitpunkt zu wiederholen.

1. Aufbereitung des Datensatzes und Vorverarbeitung [Ba+98]. Aus der UAM-Dokumentation wird ein Datensatz erstellt. Der Datensatz enthält mehrere Spalten mit Kennzahlen und als Reihen die Anwendungen. Im Rahmen der Vorverarbeitung ist eine Standardisierung durchzuführen.
2. Initialisierung der SOM unter Vorgabe von Netzwerktopologie, Lernrate und weiteren Parametern [Ek+02]
3. Konstruktion der SOM Map. Nach Beendigung und dem Abbruch des Lernens ist durch den Analysten auf Basis des quadratischen Fehlers und der Interpretation der Visualisierung zu entscheiden, ob die Ergebnisse zufriedenstellend sind [Ve+00]. Die Trainingsläufe sind zu dokumentieren.
4. Auswahl einer Map, Identifikation der Cluster und Interpretation der Ergebnisse. Die finale Map mit den Clusterergebnissen wird den relevanten Stakeholdern präsentiert. Es werden Handlungsempfehlungen formuliert.

3 Beispiel und Training der SOM-Map

In diesem Kapitel wird an einem Beispiel erläutert, wie mittels der SOM ein Clustering erfolgen kann. Für das Training und die Erzeugung der Self-Organizing Maps wurde das Programm SOM Toolbox 2.0 verwendet (vgl. [Ve+00]). Die SOM Toolbox ist ein kostenfrei verfügbares Paket für Matlab und stammt vom Adaptive Informatics Research Center in Helsinki (vgl. [AIR11]).

3.1 Einführung in das Anwendungsbeispiel

Im Mittelpunkt steht die fiktive Anwendungslandschaft eines diversifizierten Finanzdienstleisters. Betrachtet wird hier eine internationale Landesgesellschaft mit 150 Anwendungen. Für vergleichbare Größenwerte wird auf empirische Studien von Aier et al. [Ai+09] verwiesen. Die Anwendungslandschaft selbst ist über Jahre hinweg gewachsen und erscheint aus heutiger Sicht als unangemessen kostenintensiv und wenig flexibel. Das IT-Management verlangt eine Analyse über die *Komplexität* und die *Kosten* der Anwendungen in seiner Anwendungslandschaft. Als Ausgangspunkt wird auf die bestehende UAM-Dokumentation zurückgegriffen. Diese besteht aus den Anwendungen sowie den zugeordneten Funktionen und Schnittstellen (vgl. das Metamodell aus Abbildung 4). Als erster Schritt der Analyse soll ein Clustering der Anwendungen vorgenommen werden. Es wird das in Abschnitt 2.3 beschriebene Vorgehen verwendet.

ISBN 978-3-88579-286-4

3.2 Training der SOM-Map

In einem ersten Schritt werden die Daten aktualisiert und Lücken in der Dokumentation (z.B. unvollständige Funktionen und Schnittstellen) geschlossen. Da im gewählten Anwendungsbeispiel keine unvollständigen oder fehlerhaften Werte vorlagen, wird keine Einschätzung zum Verhalten der SOM im Umgang mit diesem Fall gegeben. Auf dieser Basis erfolgt eine Messung der Kennzahlenwerte. Der entstehende Datensatz bildet die Grundlage für die Initialisierung der SOM[1]. Beispielhaft ist das Datenformat in Tabelle 1 abgebildet.

Tabelle 1: Datenformat der Ausgangsdaten (beispielhaft)

	Funktionale Abhängigkeit	Interface Komplexität	Wartungskosten (in 100.000 EUR)
Anwendung 1	10	5	49
Anwendung 2	23	7	15
Anwendung 3	54	96	52
…	…	…	…

Als Trainingsdaten wird ein Sample von 20 Anwendungen verwendet. Hierfür wurde ein einfaches Random-Sampling angewendet. Die verbleibenden 130 Anwendungen verbleiben als Zieldaten und werden nachfolgend nicht weiter betrachtet. Im Bezug auf das Problem der Sample-Größe sind statistische Überlegungen zur Relevanz der Entscheidung, der Art der Analyse, Ressourcenbeschränkungen und weiteren Punkten einzubeziehen. Speziell im Bezug auf die SOM sollten auch Überlegungen zur *Lesbarkeit*[2] und *Interpretierbarkeit* der entstehenden Visualisierungen (z.B. U-Matrix, Hauptkomponentenanalyse) mitbedacht werden.

Nach dem Laden des Datensatzes in die Toolbox ist eine *Standardisierung* vorzunehmen, da höherskalierte Werte sonst in der entstehenden SOM überrepresentiert werden [Ve+00]. Dies ist notwendig, da der SOM-Algorithmus auf Grundlage der *euklidischen Distanz* arbeitet und sonst höherskalierte Werte in einer Dimension fast vollständig die Karte dominieren [Ve+00]. Fehlende oder unvollständige Werte werden bereinigt und die Skalierung der Werte wird angeglichen.

Anschließend erfolgen die *Initialisierung* und das Training der Map. Insgesamt wurden ca. 50 verschiedene Maps erzeugt und verglichen.

[1] Im Beispiel wird ein randomisierter Datenvektor mit 3 Spalten (den Kennzahlen) und 150 Reihen verwendet. Die Zufallsvariablen wurden in drei Bereichen (Cluster-A, Cluster-B, Cluster-C) zwischen 0 – 50, 51 bis 100 und 101 bis 150 erzeugt, um so die Cluster anschaulich zu machen. Cluster-D hat für die ersten Dimension (Funktionen) Werte zwischen 0-50 und für die anderen Dimensionen (Interfaces, Kosten) Werte zw. 100-150.
[2] Auf der U-Matrix leidet bei Darstellung mehrerer Datenelemente auf einem Feld sonst die Lesbarkeit. Dies ist zwar keine Clustering-Beschränkung, sollte aber beim Sampling berücksichtigt werden.

Die Auswahl einer Map erfolgte anhand des kleinsten quadratischen Fehlers und der Erkennbarkeit der Cluster auf der resultierenden Map. Dieses Abweichungsmaß erlaubt eine Beurteilung der Güte der Ergebnisse, ist aber durch die Interpretation des Analysten zu evaluieren. Im Bezug auf die Wahl der Parametereinstellungen konnten keine systematischen Unterschiede identifiziert werden. Dies ist mit der Struktur des Datensatzes zu erklären. Die Anzahl der Trainingsdurchläufe erwies sich im Beispiel als unkritisch. Er wurden 500 Iterationen eingestellt, was mit Empfehlungen in der Literatur korrespondiert [Ko95].

Die finale Map hatte eine Größe von 7 x 11 Neuronen. Zum einen ließen sich auf der gewählten U-Matrix Clustergrenzen in Form von „Wasserscheiden“ farblich gut erkennen und weiterhin war die Größe der Map für die Lesbarkeit der Elemente des Samples ausreichend groß.

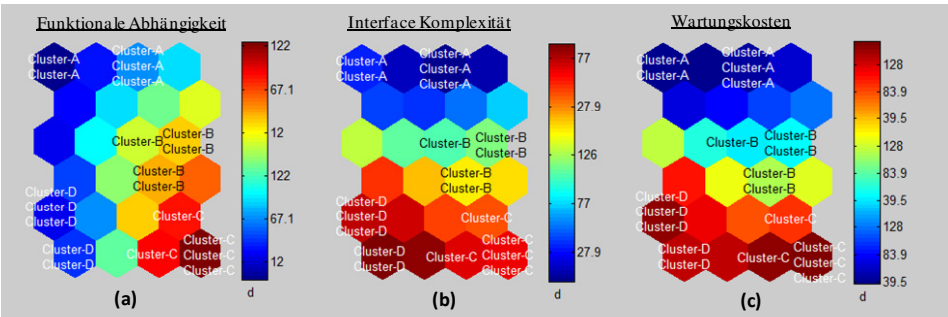

Abbildung 5: Feature Plan Map für die einzelnen Indikatoren

Abbildung 5 zeigt die Feature Plan Maps für jeden der einzelnen Indikatoren. Anhand der Farbgebung lassen sich die Unterschiede in den Merkmalsausprägungen erkennen. Interessant ist, das in Abbildung 5 (a) die Anwendungen oben und unten links jeweils niedrige Werte für die *funktionale Abhängigkeit* aufweisen. Bei den anderen Indikatoren (Interface Komplexität, Kosten) haben diese Anwendungen jeweils aber hohe Werte.

3.3 Cluster-Definition

Anhand der Analyse der endgültigen *U-Matrix* (Abbildung 6a), der Projektion auf die Hauptkomponten (Abbildung 7) und der zu Grunde liegenden *Feature Plan Maps* (Abbildung 5) wurde das in Abbildung 6b gezeigte Clustering abgeleitet. Die resultierenden Cluster können wie folgt charakterisiert werden:

- **Cluster A** zeigt Anwendungen, die für alle drei Indikatoren niedrige Werte aufweisen. Diese Anwendungen besitzen niedrige Komplexität und Kosten.
- **Cluster B** zeigt Anwendungen mit mittleren Werten die Indikatoren.
- **Cluster C** zeigt Anwendungen mit hohen Werten für Komplexität und Kosten.
- **Cluster D** zeigt Anwendungen, die hohe Werte für Kosten und Interface Komplexität aufweisen, deren funktionale Abhängigkeit aber gering ist.

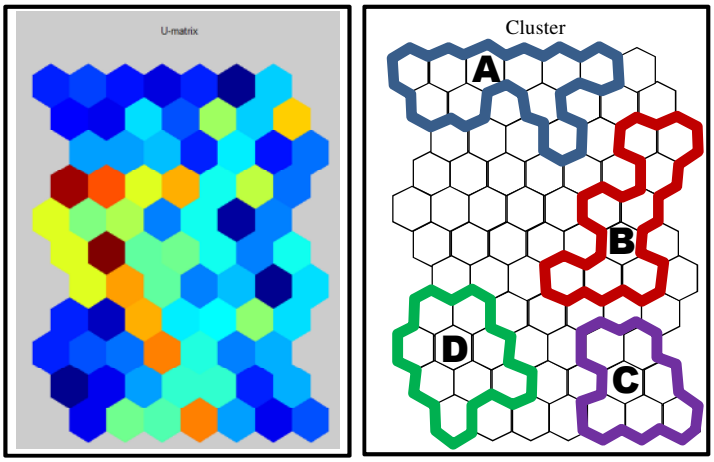

Abbildung 6: (a) finale U-Matrix und (b) identifizierte Cluster auf der Map

Tabelle 1: Charakteristika der identifizierten Cluster auf der Karte

	Cluster A	Cluster B	Cluster C	Cluster D
Funktionale Abhängigkeit	**Gering**	**Mittel**	**Hoch**	**Gering**
Interface Komplexität	**Gering**	**Mittel**	**Hoch**	**Hoch**
Wartungskosten	**Gering**	**Mittel**	**Hoch**	**Hoch**

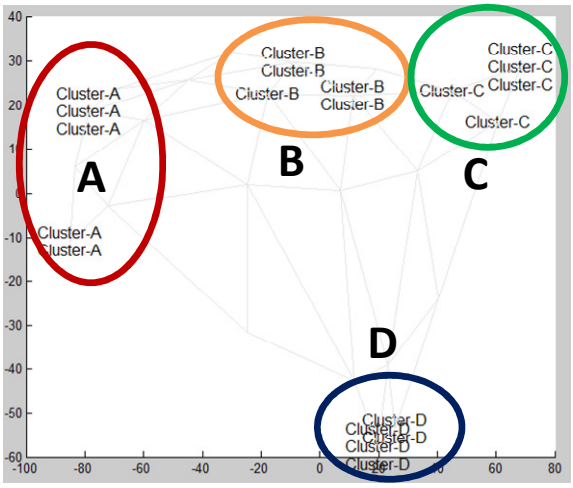

Abbildung 7: Projektion der SOM auf die Hauptkomponenten inkl. identifizierter Cluster

3.4 Interpretation und Bewertung der Ergebnisse

Anhand der Trainingsdaten im Datensatz wurden vier Cluster identifiziert (vgl. Abbildung 6b). Einige Anwendungen von Cluster B und Cluster C liegen eng beieinander, was es dem Analyst schwer macht, diese Cluster genau zu unterscheiden. Die Hauptkomponentenanalyse (vgl. Abbildung 7) erleichtet die Abgrenzung.

Für die Anwendungen in Cluster C und D wird eine vertiefte Prüfung eingeleitet. Insbesondere für die Anwendungen in Cluster D soll die Prüfung erklären, wie die widersprüchlichen Ergebnisse bei der Interface Komplexität und den Wartungskosten im Gegensatz zu der gleichzeitig niedrigen funktionalen Abhängigkeit zu Stande kommen.

4 Erfahrungen und praktische Implikationen

Es wurde illustriert, wie die Self-Organizing Map für das Clustering von Anwendungen in einem Portfolio verwendet werden kann. Dazu werden mehrere Kennzahlen verwendet, um mittels der SOM eine Gruppierung der Anwendungen vorzunehmen. Anhand der Clusterzugehörigkeit ließen sich vergleichende Aussagen über den Zustand der Anwendungen treffen. Diese aggregierten Aussagen beziehen mehrere Merkmale über die Komplexität und die Kosten der Anwendungen ein. So lassen sich bisher unbekannte Muster des Datensatzes erkennen.

Der *Nutzen* der Anwendung der SOM-Technik für die Aufgabe können folgendermaßen zusammengefasst werden: (1) Durch die Vorkategorisierung mittels der SOM sind v.a. bei großen Datenmengen Zeit- und Ressourcenersparnisse zu erwarten. Dies konnte bereits mittels des Beispiels illustriert werden. (2) Die Technik ist für den Analysten schnell erlernbar. Auch ohne vertiefte Methodenkenntnisse lassen sich interpretierbare Ergebnisse erzielen. Die erzielten Ergebnisse durch die Standard-Parametereinstellungen konnten durch Variation nicht systematisch verbessert werden. Andererseits besteht hierdurch für den Analysten die Möglichkeit sein Wissen in den Prozess einzubringen und z.B. die Größe der Map flexibel anzupassen. (3) Mit der U-Matrix stellt die SOM eine leistungsfähige graphische Darstellungstechnik als Analyse- und Interpretationshilfe zur Verfügung.

Folgende *Limitationen* wurden festgestellt: (1) Die Identifikation der Cluster ist nicht vorgegeben, sondern anhand der Visualisierungsergebnisse durch den Analysten vozunehmen. Da es sich bei der SOM um eine *heuristische* Technik handelt, können zwei trainierte Maps können abweichen. Die Ergebnisse des Clusterings sind deshalb vor dem Hintergrund von Expertenwissen zu evaluieren. In welcher Art und Weise die Experten in den Prozess eingebunden werden können, sollte empirisch weiter untersucht werden.

(2) Die Erkennung der Cluster anhand der *U-Matrix* (Abbildung 6a) erfordert eine gewisse Übung. Dies korrespondiert mit dem von Lehman et al. [Le10] genannten Nachteil reduktiver Clustering-Methoden der schwereren *Interpretierbarkeit* gegenüber nichtreduktiven Methoden. Durch die Kombination mit alternativen Cluster- und Visualisierungsmethoden kann diese Einschränkung aber gemildert werden. Die auf Grundlage der SOM erzeugte Hauptkomponentenanalyse lieferte eine visuell deutliche Abgrenzung der im Datensatz vorhandenen Cluster (vgl. Abbildung **7**). Diese und weitere Visualisierung lassen sich auf Basis der trainierten SOM in der verwendeten Toolbox erzeugen. Dennoch sollten in weiteren empirischen Arbeiten systematisch die Vor- und Nachteile der SOM im Bezug auf die Interpretierbarkeit untersucht werden. (3) Für den Nutzer vorteilhaft, in dem verwendeten Paket aber leider nicht möglich, wäre außerdem eine stärkere Nutzerinteraktion im Programm inklusive eines „Drill-Downs“.

Als Limitation in dem gewählten Vorgehen ist anzumerken, dass es sich bei der Untersuchung lediglich um eine zeitpunktbezogene Betrachtung handelt. In weiteren Arbeiten sollte untersucht werden, wie sich die SOM verhält, wenn das Clustering *wiederholt* durchgeführt wird (z.B. vierteljährlich oder jährlich). Dies umfasst auch Fragen der Reliabilität bei wiederholter Anwendung der SOM. In der Studie von Eklund et al. [Ek+02] wird durch eine Verwendung der gleichen SOM-Map bei einer mehrjährigen Studie sichergestellt, dass die Cluster auf derselben Position auf der Map erscheinen, was die Interpretierbarkeit der Ergebnisse erhöht [Ek+02]. Dies ermöglicht eine Verfolgung der Entwicklung im Zeitverlauf. Eine Übertragung auf den hier beschriebenen Anwendungsbereich wäre zu untersuchen.

5 Fazit

Die vorliegende Arbeit bringt zwei Ansätze zusammen. Den Ausgangspunkt stellen Kennzahlen über Anwendungen dar, die z.T. anhand des Beziehungswissens aus UAM-Modellen erhoben werden. Es entstehen multidimensionale Datensätze, die sich nur schwer konventionell erschließen lassen. Die SOM wird hier als Clustering-Methode und damit als Steuerungsinstrument für das IT-Management eingesetzt. Gruppen von Elementen mit ähnlichen Eigenschaften lassen sich so erkennen. Im Sinne eine Portfoliomanagements lassen sich für die einzelnen Gruppen Steuerungsmaßnahmen ableiten. Subjektive Einflüsse bei der Bewertung werden so auch bei großen Datenmengen verringert und den Analysten wird ein Mittel in die Hand gegeben, um bisher unbekannte Muster in den Daten zu erkennen. Die SOM ist heute nicht in bestehende Werkzeuge für das Architekturmanagement integriert. Die Technik steht so bisher de-facto nur einem kleinen Anwenderkreis zur Verfügung. Die Potentiale der SOM für die praktische Anwendung sollten aufbauend auf den vorgestellten Ergebnissen untersucht werden. Insbesondere ist dazu eine enge Zusammenarbeit zwischen Praxisunternehmen, Werkzeugherstellern und Forschung erforderlich.

Für weitere Forschungsarbeiten sehe ich mehrere Anknüpfungspunkte. *Erstens* sollte der Vergleich der SOM mit alternativen Clustering-Methoden für das Anwendungsgebiet intensiviert werden. Bereits vorliegende Vergleiche verschiedener Methoden (vgl. [Ba+05], [Ab07], [KK01], [SS04], [Tr08]) sollten dafür die Grundlage bilden.

ISBN 978-3-88579-286-4

Zweitens sollte die Entwicklung *valider, steuerungsrelevanter* Kennzahlen im Anwendungsgebiet als Grundlage des Clusterings vorangetrieben werden. Diese sollten möglichst leicht interpretierbar und automatisiert anhand des Beziehungswissens und der Informationen in vorhandenen UAM-Werkzeugen erhebbar sein. Außerdem sollten diese zu bestehenden Vorschlägen aus dem IT-Management (vgl. u.a. [Ma05a], [Ma05b]) und dem IT-Controlling (vgl. u.a. [Kü06], [Kü10]) kompatibel sein sollten.

Die dritte Herausforderung liegt in einer empirischen Überprüfung der Anwendbarkeit der vorgestellten Methode. Hier ergeben sich eine Reihe interessanter Fragen für weitere Arbeiten: Wie einfach lassen sich Cluster im organisatorischen Einsatz gezielt erkennen und interpretieren? Wie können die Ergebnisse zum Benchmarking genutzt werden? Außerdem ist eine vertiefte Betrachtung der zeitlichen Dimension interessant. Zunächst ließen sich einzelne Zeitpunkte komperativ-statisch gegenüberstellen, um ein Benchmarking vorzunehmen. Ein weiterer Ansatz für zukünftige Forschungen stellt die Simulation des dynamischen Verhaltens einer Anwendungslandschaft über den Zeitverlauf dar.

Literaturverzeichnis

[Ab07] Abbas, O.A.: Comparisons Between Data Clustering Algorithms, in The International Arab Journal of Information Technology (5, 3, 2008).

[Ai+09] Aier, S.; Buckl, S.; Franke, U.; Gleichauf, B.; Johnson, P., Närman, P.; Schweda, C.M.; Ullberg, J.: A Survival Analysis of Application Life Spans based on Enterprise Architecture Models. In: 3rd International Workshop on Enterprise Modelling and Information Systems Architectures, Ulm, Germany, p. 141-154, 2009.

[AIR11] Adaptive Informatics Research Center: 7718 works that have been based on the Self-Organizing Map (SOM) method developed by Kohonen. 2011. Im Internet unter: http://www.cis.hut.fi/research/som-bibl/ Letzter Zugriff: 24.03.2011.

[ARW08] Aier, S.; Riege, C.; Winter, R.: Unternehmensarchitektur – Literaturüberblick und Stand der Praxis. *WIRTSCHAFTSINFORMATIK* 50, no. 4 (9, 2008) 292-304.

[AW08] Aier, S.; Winter, R.: Virtuelle Entkopplung von fachlichen und IT-Strukturen für das IT/Business Alignment – Grundlagen, Architekturgestaltung und Umsetzung am Beispiel der Domänenbildung.*WIRTSCHAFTSINFORMATIK* 51, no. 2 (12, 2008) 175-191.

[Ba+05] Bacao, F.; Lobo, V.; Painho, M.: Self-organizing Maps as Substitutes for K-Means Clustering. In ICCS 2005, LNCS 3516, pp. 467-483, 2005.

[Ba+98] Back, B.; Sere, K.; Vanharanta, H.: Managing complexity in large data bases using self-organizing maps," *Accounting, Management and Information Technologies* 8, no. 4 (Oktober 1998): 191-210.

[De09] Dern, G.: Management von IT-Architekturen. Leitlinien für die Ausrichtung, Planung und Gestaltung von Informationssystemen. 3., durchges. Wiesbaden: Vieweg+Teubner (Praxis). 2009.

[Du10] Duggan, J.; Application Portfolio Management: TIME for the Application Masses, 2010. Gartner. Online verfügbar unter http://www.gartner.com

[EFK11] Eckert, K., Fürstenau, D., Kirchner, L . SOA-Transformation von Anwendungslandschaften. Unterstützung durch Unternehmensarchitekturmanagement. In: Praxis der Wirtschaftsinformatik - Band 278 (2), April 2011.

[Ek+02] Eklund, T.; Back, B.; Vanharanta, H.: Assessing the feasibility of self-organizing maps for data mining financial information, in *Proceedings of the Xth European Conference on Information Systems (ECIS 2002)*, 2002.

[EV08] Engels, G.; Voß, M.: Quasar Enterprise. In: Informatik-Spektrum (31, 2008) 548-555.

[En+08] Engels, G.; Hess, A.; Humm, B.; Juwig, O.: Quasar Enterprise. Anwendungslandschaften serviceorientiert gestalten. 1. Aufl. Heidelberg: Dpunkt-Verl. 2008.

[He+04] Hevner, A.R.; March, S.T.; Park, S.; Ram, S.: Design Science in Information Systems Research In: MIS Quarterly 28 (2004) 1, S. 75–105.

[Ho+08] Holschke, O.; Närman, P.; Rocha Flores, W.; Eriksson, E.; Schönherr, M: Using Enterprise Architecture Models and Bayesian Belief Networks for Failure Impact Analysis. Trends in Enterprise Architecture Research (TEAR) Workshop, Int'l Conference on Service-Oriented Computing (ICSOC 2008), Dec. 1-5, 2008, Sydney, Australia, Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg.

[JS11] Jardin, P.; Séverin, E.: Predicting corporate bankruptcy using a self-organizing map: An empirical study to improve the forecasting horizon of a financial failure model. In: Decision Support Systems (Volume 51, Issue 3, June 2011) 701-711.

[Ka97] Kaski, S.: Data exploration using self-organizing maps. 1997. PhD thesis, Helsinki University of Technology, 1997. Acta Polytechnica Scandinavia: Mathematics, Computing and Management in Engineering, 82.

[Ke06] Keller, W.: IT-Unternehmensarchitektur: Von der Geschäftsstrategie zur optimalen IT-Unterstützung, 1st Aufl. Dpunkt Verlag; 2006.

[KK01] Kiang, M.Y.; Kumar, A.: An Evaluation of Self-Organizing Map Networks as a Robust Alternative to Factor Analysis in Data Mining Applications,*INFORMATION SYSTEMS RESEARCH* 12, no. 2. Juni 1, 2001: 177-194.

[Ko95] Kohonen, T.: Self Organising Maps, 2. Aufl. Springer, Heidelberg, 1995.

[Ko+96] Kohonen, T.; Hynninen, J.; Kangas, J.; Laaksonen, J: SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996.

[Kü06] Kütz, M.: IT-Steuerung mit Kennzahlensystemen, 1st Aufl. (Dpunkt Verlag, 2006).

[Kü10] Kütz, M.: Kennzahlen in der IT: Werkzeuge für Controlling und Management, 4. Aufl. Dpunkt Verlag; 2010.

[Le10] Lehmann, D.J.; Albuquerque, G.; Eisemann, M.; Tatu, A.; Keim, D.A.; Schumann, H.; Magnor, M.A.; Theisel, H.: Visualisierung und Analyse multidimensionaler Datensätze. Informatik Spektrum (2010) 589-600.

[Ma05a] Masak, D.: Moderne Enterprise Architekturen., Springer, Berlin; 2005.

[Ma05b] Masak, D.: Legacysoftware: Das lange Leben der Altsysteme, Springer, Berlin; 2005.

[Ma06] Masak, D.: IT-Alignment. IT-Architektur und Organisation. Springer, Berlin, Heidelberg; 2006.

[Ma08a] Matthes, F.; Buckl, S.; Leitel, J.; Schweda, C. M.: Enterprise Architecture Management Tool Survey 2008. TU München, Chair for Informatics 19 (sebis), Germany, 2008.

[Ma08b] Matthes, F.: Softwarekartographie, *Informatik-Spektrum* 31, no. 6 (10, 2008): 527-536.

[Tr08] Trosset, M.W.: Representing Clusters: K-Means Clustering, Self-Organizing Maps, and Multidimensional Scaling. Technical Report 08-03. Department of Statistics. Indiana University Bloomington. February 20, 2008.

[SS04] Shahapurkar, S.S.; Sundareshan, M.K.: Comparison of Self-Organizing Maps with K-Means Clusting for Bioinformatics Applications. IEEE. 2004.

[US90] Ultsch, A.; Siemon, H. P.: Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis (1990). In Proceedings of International Neural Network Conference (INNC'90), Dordrecht, Niederlande, 1990, S. 305-308. Kluwer.

[Ve+99] Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J.: Self-organizing map in matlab: The SOM toolbox. In Proceedings of the matlab DSP conference 1999, p. 35-40, Espoo Finland, November 1999.

[Ve+00] Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J.: SOM toolbox for Matlab 5. Technical Report A57, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 2000.