Kreativität in der Informatik: Anwendungsbeispiele der innovativen Prinzipien aus TRIZ

Janine Willms, Ina Wentzlaff, Markus Specker

Abt. Lehr-/Lernsysteme, Fachbereich Informatik, Universität Oldenburg
Janine.Willms@Informatik.Uni-Oldenburg.de

Dieser Beitrag entstand im Rahmen eines Fortgeschrittenenpraktikums, das die Untersuchung und Nutzung kreativer Aspekte in der Informatik zum Thema hat. Es ist eingelagert in eine laufende Doktorarbeit zur Konzeption und prototypischen Entwicklung eines Entscheidungsunterstützungssystems für die Patentanmeldung und –prüfung mit besonderem bezug zur Informatik. Patente werden für innovative Entwicklungen erteilt, die aus der Kreativität des Entwicklers hervorgehen. Es soll untersucht werden, mit welchen Methoden eine Patentanmeldung, die nicht erfinderisch genug ist, um ein Patent zu erhalten, kreativ verändert werden kann. In diesem Beitrag wird Fragen nachgegangen, was Kreativität ist, welche Rolle die Kreativität für das Patentwesen spielt und ob man bestehende Prinzipen des kreativen Problemlösens aus der Theorie des erfinderischen Problemlösens TRIZ auch in der Informatik wiederfindet bzw. diese anwenden kann, um neue Entwicklungen voranzutreiben. Ein Hauptziel dieses Beitrags liegt somit darin, die in TRIZ definierten innovativen Prinzipien anhand von Beispielen aus dem Informatikumfeld vorzustellen.

Einleitung

Was ist eigentlich Kreativität? Ob in der Kunst, Musik, Architektur, Schriftstellerei, Technologie, wissenschaftlich Entdeckungen, aber auch im Problemlösen und Handeln – der Mensch ist fähig, kreativ zu sein, und Kreativität gilt als eine positive Eigenschaft. Eine allgemein anerkannte Definition von Kreativität existiert jedoch bisher nicht. Psychologen versuchen durch Interviews mit kreativen Menschen, deren besondere Eigenschaften zu identifizieren. Andere Forscher analysieren die historische Entwicklung kreativer Personen anhand ihrer Bibliographie (Simonton, 1990) oder von ganzen Fachgebieten auf Basis der fortschreitenden Technologien (Dasgupta, 1996). Mit Hilfe von Computersimulationen wird versucht, die kognitiven Fähigkeiten kreativer Personen erklärbar zu machen (Langley et al., 1987).
Kreativität wird auch heute noch vielfach als unerklärbarer, teils mystischer Prozeß angesehen. Betrachtet man die Begriffe *Inspiration, Eingebung, Geistesblitz oder Erleuchtung*, so wird deutlich, daß Kreativität nicht als Eigenschaft des Menschen beurteilt wurde, sondern als Eingriff einer äußeren, höheren Instanz. Aus diesem Alltagsverständnis von Kreativität ergibt sich das Problem der „Definition der immer höher gesteckten Ziele“ (Kurzweil, 1993). Die Analyse und Erklärung kreativer Prozesse sowie deren Simulation durch Computerprogramme führt lediglich dazu, daß die erreichte Leistung nicht mehr als kreativ gilt, da sie nun erklärbar geworden ist und „einfachen Regeln und Mustern“ folgt. Dennoch wird weiter versucht, Kreativitätsmuster zu entdecken und auch im Computer nutzbar zu machen.

Kreativität und Patente

Kreativität wird verstanden als:

> „Fähigkeit, möglichst viele verschiedenartige und ungewöhnliche (originelle) Lösungen einer Aufgabe produzieren zu können“ (Wörterbuch der Kognitionswissenschaft, Strube 1996)
>
> „Production of an idea, action, or object that is new and valued“ (MIT Encyclopedia of the Cognitive Sciences, Wilson, Keil, 1999)
>
> „the connecting and rearranging of knowledge - in the minds of people who will allow themselves to think flexibly - to generate new, often surprising ideas that others judge to be useful“ (Creativity, Innovation and Quality, Plsek 1997)

Die Definitionen für Kreativität lassen sich auf den folgenden gemeinsamen Grundtenor zurückführen: *Kreativität wird als Prozeß angesehen, dessen Resultat von der Gesellschaft als neu und ungewöhnlich angesehen wird. Dabei wird auf das bisher vorhandene Vorwissen zurückgegriffen, dieses kombiniert, verändert und erweitert.*
M. Boden (Boden, 1990) definiert psychologische Kreativität (P) als Leistungen, deren Ergebnisse für einen bestimmten Menschen persönlich fundamental neu sind. Das Ergebnis ist außerdem historisch kreativ (H), wenn es bezogen auf die gesamte menschliche Geschichte fundamental neu ist. Dasgupta erweitert diese Definition und trennt die fundamentale Neuheit in Neuheit (N) und Originalität (O). Dadurch ergibt sich eine Vier-Felder-Matrix. Die vier Kreativitätsarten von Dasgupta spiegeln die Phasen des Patentierungsprozesses wider. Ein Produkt des menschlichen Denkens, das PO-kreativ ist, kann als Ausgangspunkt für eine Patentanmeldung dienen. Im Patentamt wird dann geprüft, ob dieses Produkt auch bezogen auf die menschliche Rasse neu (HN-kreativ) und erfinderisch (HO-kreativ) ist.

	Neuheit (N)	Originalität (O)
Psychologische Kreativität (P)	PN – Kreativität Es wird entwickelt...	PO – Kreativität Das Ergebnis erscheint dem Entwickler als patentwürdig – er meldet ein Patent an...
Historische Kreativität (H)	HN – Kreativität Das Patentamt prüft die Anmeldung auf Neuheit bezogen auf den Stand der Technik...	HO – Kreativität Das Patentamt prüft die Anmeldung auf erfinderische Tätigkeit bezogen auf den Stand der Technik...

Fig. 1. Bezug der Kreativitätsarten nach Dasgupta zum Patentwesen

Der eigentliche Begriff der Kreativität wird im Patentwesen nicht genutzt. Patente werden für Erfindungen erteilt, die sich laut §4 PatG durch Neuheit, erfinderische Höhe und gewerbliche Anwendbarkeit auszeichnen, das heißt, sie beschreiben neue, vorteilhafte Lösungen für Problemstellungen, die einem Durchschnittsfachmann beim Sichten des Standes der Technik nicht als naheliegend erscheinen. Das heißt jedoch nicht, daß die Erfindung grundsätzlich neuartig sein muß und nicht aus dem Stand der Technik heraus entwickelt worden sein darf. Nach der Definition von Dasgupta wird durch die Bewertung, die das Patentamt vornimmt, einem Anmeldegegenstand

Kreativität bescheinigt, bzw. der Prozeß, der zu dem Gegenstand geführt hat, als kreative Leistung anerkannt.
Kategorisiert man Patente gemäß ihrer erfindungsstrukturellen Merkmale so ergeben sich folgende Erfindungsklassen (Witte, Vollrath, 1997):

- **Übertragungserfindungen**
- **Anwendungs- oder Verwendungserfindungen**
- **Auswahlerfindungen**
- **Kombinationserfindungen**
- **Fortlassungserfindungen**
- Die **Überwindung eines fachlichen Vorurteils** als Metakategorie, da für jede der vorigen Erfindungsarten möglicherweise eine Überwindung eines fachlichen Vorurteil notwendig war. Sie wird explizit aufgeführt, da sie das Urteil des Prüfers über die erfinderische Tätigkeit positiv beeinflußt.

Es stellt sich die Frage, inwiefern der Begriff der Kreativität und der der erfinderischen Höhe des Patentwesens in Einklang zu bringen sind. Der Anspruch an die erfinderische Tätigkeit des Patentwesens ist allgemein niedriger als das allgemeine Verständnis kreativer Leistungen. Bezogen auf Dasguptas Definition von Kreativität ergibt sich der endgültige kreative Charakter jedoch gerade durch die Beurteilung des Anmeldegegenstands durch das Patentamt.

Unterstützung kreativer und konstruktiver Prozesse des Menschen

Es gibt eine große Anzahl von Techniken, die die menschliche Kreativität fördern sollen. Da fehlende Motivation als hinderlich für kreative Leistungen angesehen wird, muß für ein angenehmes Umfeld und Interesse an der Fragestellung gesorgt werden. Durch die Bildung von Teams können verschiedene Sichtweisen in die Bemühung ein kreatives Ergebnis zu erreichen mit einfließen. Randomisierungsmethoden (Brainstorming, Reizwortanalyse)beruhen auf dem Abbau von sogenannen Denkblockaden (psychological inertia (Kowalick, 1998)). Neben solchen Motivations- und Organisationstechniken wird versucht, durch Fokussierungstechniken und festgelegten Vorgehensmodellen (Morphologische Analyse, Quality Function Deployment (QFD), Failure Modes and Effects Analysis (FMEA)) das vorhandene Wissen zu strukturieren, zu visualisieren und neues Wissen systematisch zu erheben. Hierbei können Computerprogramme eine wertvolle Hilfestellung bieten.
Eine neuere Form der Kreativitätsförderung sind Systeme auf Basis von Änderungsmustern und -methoden, die aus der Analyse historischer Entwicklungen gewonnen wurden. Solche Ideendatenbanken wurden aufbauend auf der Theorie des erfinderischen Problemlösens (russisches Akronym TRIZ) (Terninko, 1998) zum Beispiel in die Invention Machine integriert.
TRIZ wurde in den 50er Jahren von Altshuller, einem Angestellten des russischen Patentamtes entwickelt, um den kreativen Prozeß, der zu Innovationen führt, zu erklären und darauf aufbauend Erfindern helfen zu können. TRIZ beinhaltet Methoden zur Strukturierung von Systemen, Wissen darüber, wie Systeme sich im Laufe der Zeit verändern und Regeln zur Anwendung von Analogien. Bei der Analyse von Patenten entdeckte Altshuller 40 Grundprinzipien (Alshuller, 1998), mit deren Hilfe man ein bestehendes System verändern kann. Diese basieren wiederum auf der Veränderung von technischen und physikalischen Parametern, wie zum Beispiel Gewicht eines Objektes oder die Meßgenauigkeit eines Verfahrens.

Vielfach ist eine erfinderische Lösung eines Problems die Überwindung eine Konfliktsituation zwischen zwei Parametern. Der eine Parameter soll verändert werden, beeinflußt damit aber auch einen anderen Parameter, der eigentlich konstant gehalten werden soll. Als typisches Beispiel mag der Widerspruch zwischen Speicherplatz- und Zeitbedarf einer Funktion gelten. Das Ergebnis kann schnell geliefert werden, wenn es bereits im Speicher oder einer Datenbank vorliegt. Das ergibt jedoch gegebenenfalls einen hohen Speicherplatzbedarf. Anderseits kann der Speicherplatzbedarf reduziert werden, wenn das Ergebnis der Funktion erst berechnet wird, wenn es benötigt wird. Dieses führt jedoch zu einem erhöhten Zeitbedarf der Funktion.

Anwendung der 40 Prinzipien von TRIZ auf die Informatikdomäne

TRIZ wurde zu einer Zeit entwickelt, als die Informatik noch nicht existierte. Die analysierten Patente bezogen sich auf mechanische Vorrichtungen und den Umgang damit sowie chemische Verfahren. Es stellte sich daher die Frage, ob und wie man die Prinzipien von TRIZ auch verwenden kann, um Entwicklungen in der Informatik zu erklären bzw. voranzutreiben.
Im folgenden werden die einzelnen Prinzipien vorgestellt und anhand von Beispielen aus der Informatik erläutert. Die Übersetzung der ursprünglich englischen Begriffe wurde entsprechend (Terninko, 1998) vorgenommen. Vielfach wird deutlich, daß die Prinzipen im Hardwarebereich direkt sinnvoll einsetzbar sind, für die übrige Informatik jedoch weitreichendere Interpretationen notwendig sind. Es muß also ein neuer semantischer Rahmen geschaffen werden, um die Prinzipien in der Informatik einzusetzen. In vielen Prinzipien taucht der Begriff des „Objekts“ auf. Was ist aber das Objekt der Betrachtung in der Informatik? Wir haben versucht, durch jeweilige Ersetzung des Objektbegriffs durch Begriffe wie „Software(-architektur)“, „Hardware(-architektur)“, „Information“, „Daten“, sowie „Vorrichtung“, „System“ und „Verfahren“ Anregungen für Anwendungsbeispiele zu finden. Das Ergebnis finden Sie in den folgenden Abschnitten. Eine erweiterte Version dieses Beitrags mit Erläuterungen zu den einzelnen Stichpunkten stellen wir unter

http://lls.informatik.uni-oldenburg.de/Projekt/Kreativitaet/Informatik2000.doc

zur Verfügung.
Wir wollen verschiedene mögliche Interpretationen zur Diskussion stellen, um kreative Überlegungen über die Informatik anzuregen. Die Prinzipien und Beispiele sollten jedoch nicht als Standardlösungen angesehen werden, die ein tieferes Nachdenken über Problemsituationen nicht mehr erforderlich machen.
Prinzipien für die kein Beispiel gefunden werden konnte, werden verkleinert dargestellt.

Prinzip 1: Segmentierung
A: Zerlege ein Objekt in unabhängige Teile.
- *Adressraum im Internet*
- *Imperative vs. modulare/objektorientierte Programmierung.*

B: Mache ein Objekt zerlegbar.
- *Mounten von Festplatten unter UNIX/Linux.*
- *Aufrüsten von RAM oder weiteren Systemkomponenten.*
- *Verkettete Listen.*

C: Erhöhe den Grad der Zerlegbarkeit.
- *Back- und Frontends im Compilerbau.*
- *Programme für Multiprozessorsysteme.*

Prinzip 2: Abtrennung
A: Entfernung oder Abtrennung des störenden Objektes.
- *Einsatz von Filtern.*
- *Einsatz von Masken.*
- *One-Click-Procedures.*

B: Den notwendigen Teil bzw. wesentliche Eigenschaften alleine einsetzen
- *Stationierung/Tankstellen.*
- *virtuelle/abstrakte Funktionen, Klassen/Polymorphie.*

Prinzip 3: Örtliche Qualität
A: Übergang von homogener Struktur des Objektes oder seiner Umgebung zu einer heterogenen Struktur.
- *Symmetric vs. Massiv Parallel Processing.*
- *Konservative vs. Optimistische Simulationsverfahren/Datenbankzugriffe.*

B: Die verschiedenen Teile eines Systems sollen verschiedene Funktionen erfüllen.
- *Speicherhierarchien.*
- *Konstruktor/Destruktor.*

C: Jede Komponente eines Systems unter für sie individuell optimalen Bedingungen einsetzen.
- *Informationhiding.*

Prinzip 4: Asymmetrie
A: Ersetze symmetrische Formen durch asymmetrische.
- *Takten.*
- *Schleifen vs. Sprünge.*
- *Public Key-Verschüsselung.*

B: Erhöhe den Grad an Asymmetrie, wenn diese schon vorliegt
- *(Automatische) Schrittweitenregulierung.*
- *Von zeitdiskreten zu ereignisdiskreten Modellklassen.*

Prinzip 5: Vereinen
A: Guppiere gleichartige oder zur Zusammenarbeit bestimmte Objekte räumlich zusammen
- *Pipelining.*
- *Multiprozessorsysteme.*
- *Rechnernetze.*

B: Vereine gleichartige oder zur Zusammenarbeit bestimmte Objekte, d.h. kopple sie zeitgleich
- *Multitasking/Multiusersysteme.*

Prinzip 6: Universalität
A: Das System erfüllt mehrere unterschiedliche Funktionen, wodurch andere Systeme oder Objekte überflüssig werden.
- *Emulatoren.*
- *Videokarten im PC.*
- *Designpattern.*

Prinzip 7: Verschachtelung
A: Ein Objekt befindet sich im Inneren eines anderen Objektes, das sich ebenfalls im Inneren eines dritten befindet.
- *Rekursion.*
- *Frames in Webseiten.*

B: Ein Objekt paßt in oder durch den Hohlraum eines anderen.
- *Komprimierung.*
- *Steganographie*.

Prinzip 8: Gegengewicht
A: Das Gewicht des Objekts kann durch Kopplung an ein anderes, entsprechend tragfähiges Objekt kompensiert werden.
- *Customer/Producer-Lösungen.*

B: Das Gewicht des Objekts kann durch aerodynamische oder hydraulische Kräfte kompensiert werden.
- *Vakuum im Festplatteninneren.*

Prinzip 9: Vorgezogene Gegenaktion
A: Vor der Ausführung einer Aktion muß eine erforderliche Gegenaktion vorab ausgeführt werden.
- *Antivirenimpfung von Programmen.*
- *Initialisierung von Variablen.*

B: Muß ein Objekt in Spannung sein, dann muß vorab die Gegenspannung erzeugt werden.

Prinzip 10: Vorgezogene Aktion
A: Führe die notwendige Aktion - teilweise oder ganz - im voraus aus.
- *Automatisches Zwischen-)Speichern.*
- *Look ahead caching.*

B: Ordne Objekte so an, daß sie ohne Zeitverlust vom richtigen Ort aus arbeiten können.
- *Trojanische Pferde.*

Prinzip 11: Vorbeugemaßnahmen
A: Kompensiere die schlechte Zuverlässigkeit eines Systems durch vorher ergriffene Gegenmaßnahmen.
- *Exception(handling).*
- *Sicherheitshardware.*
- *Backups.*
- *Sicherheitsmarkierungen.*
- *Deadlock-Detection.*

Prinzip 12: Äquipotential
A: Verändere die Bedingungen so, daß das Objekt mit konstantem Energiepotential arbeiten kann, also weder angehoben, noch abgesenkt werden muß.
- *Zeiger in Programmiersprachen.*

Prinzip 13: Umkehrung
A: Implementiere anstelle der durch Spezifikation diktierten Aktion genau die gegenteilige Aktion.
- *positive vs. negative Gatter.*

B: Mache ein unbewegliches Objekt beweglich oder ein bewegliches unbeweglich.
- *Variablen/Konstanten.*

C: Stelle das System „auf den Kopf“, kehr es um.
- *Maus vs. Trackball.*
- *Top-Down- vs. Buttom-Up-Verfahren.*

Prinzip 14: Krümmung

A: Ersetze lineare Teile oder flache Oberflächen durch gebogene, kubische Strukturen druch sphärische.
- *Maus und Trackball vs. Grafiktablett*

B: Benutze Rollen, Kugeln, Spiralen. sphärische.
- *Maus und Trackball*

C: Ersetze lineare Bewegungen durch rotierende, nutze die Zentrifugalkraft.
- *Festplatte und CD vs. Band.*

Prinzip 15: Dynamisierung

A: Gestalte ein System oder dessen Umgebung so, daß es sich automatisch unter allen Betriebszuständen auf optimale Performance einstellt.
- *dynamische, flexible Bandbreite(nanpassung).*

B: Zerteile ein System in Elemente, die sich untereinander optimal arrangieren können.
- *Costumer-Setup oder benutzerspezifische Arbeitsoberflächen.*
- *Mikrooperationen.*

C: mache ein unbewegliches Objekt beweglich, verstellbar oder austauschbar.
- *CD-Wechsler oder Wechselfestplatten.*

Prinzip 16: Partielle oder überschüssige Wirkung

A: Wenn es schwierig ist, 100% einer geforderten Funktion zu erreichen, verwirkliche etwas mehr oder weniger, um so das Problem deutlich zu vereinfachen.
- *Codeoptimierung.*

Prinzip 17: Höhere Dimension

A: Umgehe Schwierigkeiten bei der Bewegung eines Objektes entlang einer Linie durch eine zweidimensionale Bewegung (in einer Ebene). Analog wird ein Bewegungsproblem in der Ebene vereinfacht durch Übergang in die dritte Dimension.
- *Holographische (3D)-Speicher.*

B: Ordne Objekte in mehreren statt in einer Ebene an.
- *Lineare Listen und Vektoren vs. (mehrdimensionale) Arrays.*
- *Nutzung von Metainterpretern*

C: Plaziere das Objekt geneigt, oder kippe es.
- *Desktop vs. Tower.*

D: Nutze Projektionen in die Nachbarschaft oder auf die Rückseite des Objektes.
- *Höhere Programmiersprachen vs. Assembler.*

Prinzip 18: (Mechanische) Schwingungen

A: Versetze ein Objekt in Schwingungen.
- *Endlosschleifen.*

B: Ozilliert das Objekt bereits, erhöhe seine Frequenz.
- *Taktfrequenz eines Rechnersystems.*

C: Benutze die Resonanzfrequenz(en).

D: Ersetze mechanische Schwingungen durch Piezovibrationen.

E: Setze Ultraschall in Verbindung mit elektromagnetischen Feldern ein.

Prinzip 19: Periodische Wirkung

A: Übertragung von kontinuierlicher zu periodischer Wirkung.

- *Time-Division Duplex.*

B: liegt bereits eine periodische Aktion vor, verändere deren Frequenz.

- *Die Clock.*

C: Benutze Pausen zwischen einzelnen Impulsen, um andere Aktionen einfügen zu können.

- *Hidden arbitration.*
- *Seti@Home-Projekt*.

Prinzip 20: Kontinuität

A: Führe eine Aktion ohne Unterbrechung aus, alle Komponenten sollen ständig mit gleichmäßiger Belastung arbeiten.

- *Schreibende Datenbankzugriffe.*

B: Schalte Leerläufe und Unterbrechungen aus.

- *Exceptions oder Interrupts.*

C: Ersetze eine „vor-und-zurück"-Bewegung durch eine rotierende.

- *Bandlaufwerk vs. Festplatte.*
- *FIFO statt LIFO.*

Prinzip 21: Überspringen

A: Führe schädliche oder gefährliche Aktionen mit sehr hoher Geschwindigkeit durch.

- *Rapid-Prototyping.*

Prinzip 22: Schädliches in Nützliches wandeln.

A: Nutze schädliche Faktoren oder Effekte – speziell aus der Umgebung – positiv aus.

- *Defragmentieren durch Überschreibung der Daten mit 0.*

B: Beseitige einen schädlichen Faktor durch Kombination mit einem anderen schädlichen Faktor.

C: Verstärke einen schädlichen Einfluß soweit, bis er aufhört, schädlich zu sein.

- *SATAN:* ein Programm, welches Sicherheitslücken in einem System aufdecken kann.

Prinzip 23: Rückkopplung

A: Führe eine Rückkopplung ein.

- *Energiesparmodus.*

B: Ist eine Rückkopplung vorhanden, ändere sie oder kehre sie um.

- *Netzwerkprotokolle:* handshake etc.

Prinzip 24: Mediatoren, Vermittler

A: Nutze ein Zwischenobjekt, um eine Aktion weiterzugeben oder auszuführen.

- *Router:* helfen bei der Paketweiterleitung.
- *Arbiter:* können z.B. den Zugriff von CPUs auf den Bus steuern.
- *Agentensysteme:* helfen z.B. bei der Suche nach bestimmten Informationen.

B: Verbinde das System zeitweise mit einem anderen, leicht zu entfernenden Objekt.

- *Lokale Variablen:* zur Speicherung temporär-interessierender Werte.
- *Design Pattern " Mediator"*.

Prinzip 25: Selbstversorgung

A: Das System soll sich selbst bedienen und Hilfs- sowie Reparaturfunktionen selbst ausführen.

- *Selbst-Diagnose.*

B: Nutze Abfall und Verlustenergie.

- *Zur sicheren Löschung von Daten, werden sie mit unsinnigen Daten überschrieben*

Prinzip 26: Prinzip 26: Kopieren

A: Benutze eine billige, einfache Kopie anstatt eines komplexen, teueren, zerbrechlichen oder schlecht handhabbaren Objektes.

- *Simulatoren/Simulation.*
- *Unix-Filesystem.*

B: Ersetze ein System oder ein Objekt durch eine optische Kopie oder Abbildung. Hierbei kann der Maßstab (vergrößern, verkleinern) verändert werden.

- *Touchscreen statt Tastatur.*

C: Werden bereits optische Kopien benutzt, dann gehe zu infrarot oder ultravioletten Abbildungen über.

Prinzip 27: Billige Kurzlebigkeit

A: Ersetze ein teures System durch ein Sortiment billiger Teile, wobei auf einige Eigenschaften (Langlebigkeit beispielsweise) verzichtet wird.

- *PIN und TAN.*
- *Zufallszahlen.*

Prinzip 28: Mechanik ersetzen

A: Ersetze ein mechanisches System durch ein optisches, akustisches oder geruchsbasierendes System.

- *CD(-ROM) statt HDD***.**

B: Benutze elektrische, magnetische oder elektromagnetische Felder.

- *HDD statt Lochkarten.*

C: Ersetze Felder: statinäre durch bewegliche, konstante durch periodische, strukturlose durch strukturierte.

D: Setze Felder in Verbindung mit ferromagnetischen Teilchen ein.

Prinzip 29: Pneumatik und Hydraulik

A: Ersetze feste, schwere Teile eines Systems durch gasförmige oder flüssige. Nutze Wasser oder Luft zum Aufpumpen, Luftkissen, hydrostatische Elemente.

- *Verwendung bio-chemischer Speicher.*

Prinzip 30: Flexible Hüllen und Filme

A: Ersetze übliche Konstruktionen durch flexible Hüllen oder dünne Filme.

- *Wrapper.*
- *Adaptive Benutzungsschnittstellen, Adaptive Portale..*

B: Isoliere ein Objekt von der Umwelt durch einen dünnen Film oder eine Membran.

- *Firewalls.*

Prinzip 31: Poröse Materialien*

A: Gestalte ein Objekt porös oder füge poröse Materialien (Einsätze, Überzüge...) zu.

- *Puffer in Rechnernetzen.*

B: Ist ein Objekt bereits porös, dann fülle die Poren mit einem vorteilhaften Stoff im voraus.

- *Halbleiter.*

Prinzip 32: Farbänderungen

A: Verändere die Farbe eines Objektes oder die der Umgebung.
- *Laser im CD-ROM-Laufwerk.*
- *Farbunterschied zwischen aktiven und nichtaktiven Anwendungen in GUIs.*

B: Verändere die Durchsichtigkeit eines Objektes oder die der Umgebung.
- *Abstrakte Datentypen.*

C: Nutze zur Beobachtung schlecht sichtbarer Objekte oder Prozesse geeignete Farbzusätze.
- *Colorierte Petri-Netze.*

D: Existieren derartige Farbzusätze bereits, setze Leuchtstoffe, lumineszente oder anderweitige markierte Substanzen ein. .
- *Blinken, Bewegung in Benutzungsoberflächen.*

Prinzip 33: Homogenität

A: Fertige interagierende Objekte aus demselben oder aus ähnlichem Material.
- *Metadaten.*

Prinzip 34: Beseitigung und Regeneration

A: Beseitige oder verwerte (ablegen, auflösen, verdampfen) diejenigen Teile des Systems, die ihre Funktion erfüllt haben oder unbrauchbar geworden sind.
- *Pakete.*

B: Stelle verbrauchte Systemteile unmittelbar – im Arbeitsvorgang – wieder her.
- *Token-Erzeugung* .

Prinzip 35: Eigenschaftsänderung

A: Ändere den Aggregatzustand eines Objektes: fest, flüssig, gasförmig, aber auch quasiflüssig oder ändere Eigenschaften wie Konzentration, Dichte, Elastizität, Temperatur.
- *Änderung der physikalischen Dichte von Daten auf Festplatten*
- *Defragmentierung*.

Prinzip 36: Phasenübergänge

A: Nutze die Effekte während des Phasenüberganges einer Substanz aus: Volumenänderung, Wärmeentwicklung oder -absorption.

Prinzip 37: Wärmeausdehnung

A: Nutze die thermische Expansion oder Kontraktion von Materialien aus.
B: Benutze Materialien mit unterschiedlichen Wärmeausdehnungskoeffizienten.

Prinzip 38: Prinzip 38: Starkes Oxidationsmittel

A: Ersetze normale Luft durch sauerstoffangereicherte Luft.
B: Ersetze angereicherte Luft durch reinen Sauerstoff.
C: Setze Luft oder Sauerstoff ionisierenden Strahlen aus.
D: Benutze Ozon.

Prinzip 39: Inertes (veraltet für: träges) Medium

A: Ersetze die übliche Umgebung durch eine inerte.
B: Führe den Prozeß im Vakuum aus.

Prinzip 40: Verbundmaterial
A: Ersetze homogene Stoffe durch Verbundmaterialien.
- *Array vs. Record.*

Zusammenfassung

Durch den hohen Abstraktionsgrad einiger Prinzipien aus TRIZ sind diese problemlos auch in der Informatik anwendbar und wurden auch bereits angewendet, wie in den vorigen Abschnitten gezeigt werden konnte. Einige Prinzipien basieren jedoch stark auf Effekten aus der Natur und sind somit in Harwarebereich, weniger aber in der substanzlosen Ideenwelt der Algorithmen und Verfahren einsetzbar.
Für die Zukunft wäre eine Analyse der informations- und softwaretechnischen Verfahren interessant, um in diesen speziellen Gebieten neue Innovationsprinzipien aufzudecken.

Referenzen

Altshuller, G., “40 Priciples: TRIZ Keys to Technical Innovation”, TRIZ Tools, Vol. 1, Worcester, MA: Technical Innovation Center, 1998.

Boden, M., „The Creative Mind”, London: Abacus, 1990.

Dasgupta, S., „Technology and Creativity”, Oxford: Oxford University Press, 1996.

Kowalick, J., „Psychological Inertia“, in: TRIZ Journal, August, 1998.

Kurzweil, R., “Das Zeitalter der künstlichen Intelligenz”, München:Carl Hanser Verlag, 1993.

Langley, P., Simon, H.A., Bradshaw, G.L, Zytkow, J.M., „Scientific Discovery“, MIT Press, 1987.

Plsek, P.E., ”Creativity, Innovation and Quality”, Milwaukee: ASQC Quality Press, 1997.

Simonton, D.K., „Scientific Genius“, Cambridge University Press, 1990.

Strube, G., “Wörterbuch der Kognitionswissenschaft”, Stuttgart: Klett-Cotta, 1996.

Terninko, J., “TRIZ – der Weg zum konkurrenzlosen Erfolgsprodukt”, R.Herb (Hrsg.), Verlag Moderne Industrie, 1998.

Wilson, R.A., Keil, F.C., “The MIT Encyclopedia of the Cognitive Sciences”, London: MIT Press, 1999.

Witte, J., Vollrath, U., “Praxis der Patent- und Gebrauchsmusteranmeldung”, C. Heymanns Verlag, Köln, 1997.