

ARBEITSPAPIERE

WORKING PAPERS

NR. 8, JULI 2011

INTERAKTION BEI DER COMPUTERGESTÜTZTEN  
GENERIERUNG VON LAYOUTS

SVEN SCHNEIDER, REINHARD KÖNIG

ISSN 2191-2416



**Sven Schneider, Reinhard König**

Interaktion bei der computergestützten Generierung von Layouts

Weimar 2011

Arbeitspapiere (Working Papers) Nr. 8

Informatik in der Architektur, Bauhaus Universität Weimar

Herausgegeben von Prof. Dr. Dirk Donath und Dr. Reinhard König

ISSN 2191-2416

Bauhaus-Universität Weimar, Professur Informatik in der Architektur

Belvederer Allee 1, 99425 Weimar

<http://infar.architektur.uni-weimar.de>

Titelbild: Jugendstil-Wendeltreppe im Hauptgebäude © Bauhaus-Universität Weimar

Redaktionelle Anmerkung:

Sven Schneider ist wissenschaftlicher Mitarbeiter an der Professur Informatik in der Architektur an der Bauhaus-Universität Weimar.

Dr. Reinhard König ist Vertretungsprofessor an der Professur Informatik in der Architektur an der Bauhaus-Universität Weimar.

Der Text ist im Rahmen des von der DFG geförderten Forschungsprojekts „KREMLAS: Entwicklung einer kreativen evolutionären Entwurfsmethode für Layoutprobleme in Architektur und Städtebau“ (DO 551/19-1) entstanden. <http://infar.architektur.uni-weimar.de/service/drupal-cms/kremlas>

# Nutzerinteraktion bei der computergestützten Generierung von Layouts

Sven Schneider, Reinhard König

sven.schneider@uni-weimar.de, reinhard.koenig@uni-weimar.de

Professur Informatik in der Architektur

Fakultät Architektur, Bauhaus-Universität Weimar, Belvederer Allee 1, 99421 Weimar, Germany

## Abstract

Das vorliegende Arbeitspapier beschäftigt sich mit der Thematik der Nutzerinteraktion bei computerbasierten generativen Systemen. Zunächst wird erläutert, warum es notwendig ist, den Nutzer eines solchen Systems in den Generierungsprozess zu involvieren. Darauf aufbauend werden Anforderungen an ein interaktives generatives System formuliert. Anhand eines Systems zur Generierung von Layouts werden Methoden diskutiert, um diesen Anforderungen gerecht zu werden. Es wird gezeigt, dass sich insbesondere evolutionäre Algorithmen für ein interaktives entwurfsunterstützendes System eignen. Es wird kurz beschrieben, wie sich Layoutprobleme durch eine evolutionäre Strategie lösen lassen. Abschließend werden Fragen bezüglich der grafischen Darstellung von Layoutlösungen und der Interaktion mit dem Dargestellten diskutiert.

**Keywords:** interaktive generative Entwurfssysteme; Entwurfswerkeugentwicklung; Evolutionäre Algorithmen

## 1. Einleitung

Mittels computerbasierter Generative Systeme können für bestimmte Problemstellungen automatisch Lösungen generiert werden. Solche Systeme eignen sich insofern zur Unterstützung des Entwurfsprozesses, da sie es erlauben, in relativ kurzer Zeit eine große Anzahl von Entwürfen zu generieren und zu überprüfen. Kriterien, die vom Menschen schlecht oder nur langsam überprüft werden können, können mithilfe Generativer Systeme effektiv in den Entwurfsprozess eingebunden werden (Eckert, Kelly, & Stacey, 1999). Bei einer komplexen Entwurfsaufgabe können beispielsweise Teilprobleme an den Computer übergeben werden, welche für diese Probleme schnell und gute Lösungen finden kann. Bei diesen Teilproblemen handelt es sich um formulierbare bzw. operationalisierbare Aspekte, welche in aller Regel die funktionalen Anforderungen an einen Entwurf darstellen.

Beim „traditionellen“ Entwerfen (gemeint ist hiermit das Entwerfen ohne computerbasierte Generative Systeme) werden in der Regel Entwurfskonzepte aufgestellt, von denen man annimmt, dass sie alle Kriterien zufriedenstellend berücksichtigen können. Die mit Hilfe einer solche „Entwurfshypothese“ entwickelten vorläufigen Lösungen werden dann im Laufe des Entwurfsprozess immer wieder überprüft und verfeinert. Voraussetzung für dieses Vorgehen ist eine wesentliche Reduktion der Komplexität einer Entwurfsaufgabe unter zur Hilfenahme von Leitbildern<sup>1</sup> (Lawson, 2006) auf einer möglichst hohen Maßstabsebene. Diese Leitbilder geben den Rahmen bzw. die Regeln für die Gestaltung auf untergeordneten Maßstabsebenen vor. Eine solche Arbeitsweise kann als Top-Down Entwurfsstrategie bezeichnet werden, wobei die funktionalen Anforderungen in der Regel den Leitbildern untergeordnet werden.

Die gestalterische Ausarbeitung einer Entwurfssaufgabe findet weitgehend unabhängig von der Optimierung funktionaler Gesichtspunkte statt. Die automatisierte Lösung von Entwurfssaufgaben durch computerbasierte Methoden findet dagegen weitgehend unabhängig von gestalterischen Absichten bzw. subjektiven Vorstellungen statt. In einem wechselseitigen Zusammenspiel von Gestaltungs- und Optimierungsmethoden liegt ein großes Potential für die Verbesserung computergestützter Entwurfsprozesse. Ein wesentlicher Aspekt hierfür ist die im vorliegenden Arbeitspapier thematisierte Interaktion des Nutzers (Entwerfers) mit dem Generativen System.

---

<sup>1</sup> Lawson spricht von „guiding images“ bzw. „guiding principles“

## 2. Notwendigkeit der Interaktion

Begeistert von den Fortschritten in der Computertechnik und künstlichen Intelligenz beschreibt Negroponte 1970 in seinem Buch *“The architecture machine”* einen intelligenten Entwurfsautomat: *„Imagine a machine that can follow your design methodology and at the same time discern and assimilate your conversational idiosyncrasies. This same machine, after observing your behavior, could build a predictive model of your conversational performance. Such a machine could then reinforce the dialogue by using the predictive model to respond to you in a manner that is in rhythm with your personal behavior and conversational idiosyncrasies.”* (Negroponte, 1970). Eine Beschreibung der Methoden, die die Umsetzung einer solchen Maschine erlauben würden, bleibt Negroponte jedoch schuldig. Die Idee einer intelligenten Entwurfsmaschine ist bis heute eine Vision. So schreibt Schön (1992) resignierend, dass Software, die ähnlich wie ein Entwerfer handeln soll, dies nur kann in einer *“highly restricted situation, a narrowly defined chunk of a design process, where the design world employed by designers can feasibly be assumed as given and fixed”*.

Warum eine vollständige Entwurfsautomatisierung nicht nur ein technisches, sondern auch konzeptionelles Problem ist, wird deutlich, wenn man sich die Charakteristika von Entwurfproblemen vergegenwärtigt. Dazu ist eine Unterscheidung zwischen operationalen und nicht-operationalen Problemen hilfreich. Ein Problem ist operational, wenn es so genau beschrieben werden kann, dass sich angeben lässt, durch welche Schritte es zu lösen ist. Dagegen sind nicht-operationale Probleme *„vage definiert, bedeutende Elemente der Aufgabestellung sind unbekannt oder nicht genau (quantitativ) erfassbar, ihr Lösungskriterium ist nicht eindeutig formuliert; der Entscheidungsprozess beschäftigt sich weniger mit der Suche nach Lösungen, sondern vielmehr mit der Konkretisierung und Abgrenzung des Problems sowie der Schließung offener Beschränkungen“* (Röpke, 1977). Rittel und Webber (1973) nennen diese Probleme aufgrund ihrer Eigenschaften auch „wicked problems“ oder bösartige bzw. verzwickte Probleme. Bei Entwurfproblemen handelt es sich in der Regel um solche nicht-operationalen Probleme (Simon, 1969). Für diese Probleme gibt es keine eindeutig richtige oder falsche Lösung. Die Lösung solcher Probleme hängt daher immer von subjektiven und kontextabhängigen Kriterien ab<sup>2</sup>. Eine vollständige Formalisierung solcher nicht-operationaler Entwurfprobleme, also ihre Überführung in operationale Fragen, würde bedeuten, *„dass der Zusammenhang zwischen den Umgebungsvariablen und der neuronalen Aktivität des Organismus vermessen wird“* (Franck & Elezkurtaj, 2002). Die

---

<sup>2</sup> Folgerichtig beschreibt Negroponte seinen Entwurfsautomaten auch als eine intelligente, denkende Maschine, die sich auf den Entwerfenden einstellt und im Dialog mit ihm Lösungen entwickelt.

umfassende empirische Erhebung der dafür notwendigen Daten erscheint trotz aller Fortschritte in den Kognitionswissenschaften kaum möglich<sup>3</sup>. Ein pragmatischer Ansatz zum Umgang mit diesem Dilemma im Kontext von computerbasierten Entwurfssystemen besteht in der Einbeziehung des Nutzers in den generativen Prozess: Während sich operationale Probleme meist sehr gut algorithmisch lösen lassen, ist man bei nicht-operationalen Problemen auf die Interpretation eines Problems durch den Menschen angewiesen. Die Einbeziehung menschlicher Fähigkeiten in ein generatives Entwurfssystem ermöglicht es, bestimmte Aspekte des Entwurfsprozesses auf operationale Probleme zu reduzieren, denn, *„all that is possible is the conversion of particular problems from ill-structured to well-structured via the one transducer that exists, namely, man“* (Ernst & Newell, 1969, p. 363).

### 3. Anforderungen an ein interaktives Entwurfssystem

Für die Konzeption von Generativen Entwurfssystemen ist es wesentlich, sich das Zusammenspiels von Entwerfer und Werkzeug (siehe Abb. 1) zu vergegenwärtigen. Dieses Zusammenspiel kann als eine Art Kreislauf aufgefasst werden (Gänshirt, 2007). Dabei benutzen Entwerfer Werkzeuge um ihre (zunächst unsichtbaren) Gedanken und Ideen zu externalisieren, sprich in Form von Artefakten (Skizzen, Modelle, Bilder, etc.) sichtbar zu machen. Diese Artefakte können in der Folge bewertet, mit den Entwurfskriterien abgeglichen und mittels verschiedener Entwurfswerkzeuge weiterentwickelt werden. Schön (1983) nennt diesen Kreislauf auch eine *„reflective conversation with the situation“*. Dabei ist zu bemerken, dass jedes Werkzeug nur einen bestimmten Raum an möglichen Operationen bietet. Diese begrenzen den Handlungsspielraum des Benutzers eines bestimmten Werkzeuges, sprich die Menge möglicher Lösungen, die mit einem Werkzeug innerhalb eines bestimmten Zeitraumes realisierbar ist. Je größer dieser Handlungsspielraum, desto uneingeschränkter ist man bei der Suche nach geeigneten Lösungen für ein Problem.

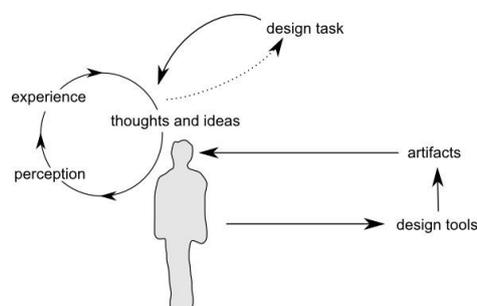


Abb. 1: Entwurfskreislauf als Zusammenspiel zwischen Mensch und Werkzeug

<sup>3</sup> So kommt Schön (1992) zu dem Schluss, dass *„practitioners of Artificial Intelligence in design world would do better to aim at producing design assistants rather than knowledge systems phenomenologically equivalent to those of designers.“*

Für die Entwicklung eines generativen Entwurfssystems zur Unterstützung von Entwurfsprozessen bedeutet dies zweierlei: zum einen muss es sich möglichst nahtlos in den beschriebenen Kreislauf (Abb. 1) einfügen. Schön (1992) spricht hier von der Berücksichtigung des elementarsten Bestandteils des Entwurfsprozesses, dem „*seeing-moving-seeing*“. Damit ist eine Fähigkeit gemeint, die es ermöglicht, auf das Wahrgenommene zu reagieren, es weiterzuentwickeln und erneut wahrzunehmen. Zum anderen ist es entscheidend, den Handlungsspielraum, den das Werkzeug bietet, möglichst groß zu halten. Die Qualität einer Software hängt also in entscheidendem Maße von ihrem Potential zum Dialog mit dem Nutzer (conversation) und ihrer Unvorherbestimmtheit (experience of surprise) ab (Schön, 1992). Im Folgenden werden vier Eigenschaften beschrieben, die ein generatives Entwurfssystem aufweisen muss, um den genannten Anforderungen gerecht zu werden.

### 3.1. Adaptivität (Flexibilität)

Da sich Wahrnehmung der Probleme und die Reaktion auf diese im Entwurfsprozess parallel entwickeln (Schön, 1983), können Entwurfsprobleme nicht ad hoc mittels konkreten, vorgefertigten Formeln abgebildet werden. Im Gegensatz zu „*presented problem situations*“ hat man es beim Entwerfen mit „*discovered problem situations*“ zu tun (Getzels & Csikszentmihalyi, 1967). Entwurfsprobleme sind also nicht a priori vorgegeben, sondern werden Schritt für Schritt während des Entwurfsprozesses definiert.

Ein generatives System muss es daher erlauben, während des Suchprozesses neue Anforderungen für ein Entwurfsproblem definieren bzw. bestehende Anforderungen ändern zu können. Adaptiv bedeutet folglich, dass die Problemdefinition möglichst flexibel anpassbar ist. Voraussetzung für eine solche Adaptivität die im Folgenden beschriebene Zirkularität.

### 3.2. Zirkularität (Kontinuität)

Entwerfen ist ein iterativer Prozess. Jede gefundene Lösung kann aus einer anderen Perspektive wieder neue Probleme bergen, die in weiteren Entwurfsiterationen behandelt werden müssen. Dazu müssen oft neue Anforderungen für das Problem formuliert werden (siehe 3.1). Die Reihenfolge für die Definition der Anforderungen ist jedoch nicht festgeschrieben. Der fortwährende Anpassungsprozess folgt keinem vordefinierten Schema.

Entscheidend für das Generative System ist es daher, dass dieses kontinuierlich arbeitet. Es gibt also keinen konkreten Anfangs- oder Endpunkt für den Lösungsprozess, sondern es befindet sich immer „in Bereitschaft“, um auf sich ändernde Anforderungen (siehe 3.1) reagieren zu können.

### 3.3. Explorativität

Generative Systeme können Lösungen erzeugen, welche bestimmten operationalen Kriterien entsprechen. Diese Kriterien und deren Gewichtung können bei jedem Entwerfer bzw. bei jedem Entwurf variieren. Entscheidend ist dafür, dass die Bewertungsfunktion des Generativen Systems flexibel definiert werden kann.

Hinsichtlich sich widersprechender Kriterien, die sich während eines Entwurfsprozesses ergeben, ist es wichtig, dass das Generative System ein möglichst breites Spektrum an gleichwertigen Kompromisslösungen anbietet. Das bedeutet, dass das System Lösungen anbietet, die möglichst gleichmäßig über die sogenannte Paretofront<sup>4</sup> (auf der alle Kompromisslösungen liegen), verteilt sind. Dadurch wird sichergestellt, dass alle relevanten Kompromisse angegeben werden. Die Auswahl zwischen den gleichberechtigten Kompromisslösungen muss durch einen Nutzer erfolgen, sofern keine weiteren Kriterien eingeführt werden.

Zu bemerken ist, dass die Suche nach Kompromisslösungen nur in einem gewissen Rahmen geschehen kann. Dieser Rahmen ist definiert durch die im Generierungsalgorithmus festgeschriebenen Regeln zur Erzeugung einer Lösung und der Definition des Problems. Damit sich die Lösungssuche nicht im Überprüfen von Trivialfällen erschöpft, sollten einerseits die Generierungsregeln ein möglichst großes Spektrum an Lösungen zulassen, andererseits müssen zahlreiche Entwurfskriterien formuliert und gegebenenfalls miteinander kombiniert werden können.

### 3.4. Unmittelbarkeit

Während des Entwurfsprozesses gibt es sowohl Phasen in denen der Entwerfende die erarbeitete Lösung in aller Ruhe reflektiert, als auch Phasen in denen er in hoher Geschwindigkeit verschiedene Dinge gleichzeitig zu organisieren versucht (Lawson, 2005). Der Entwerfer muss in den entscheidenden Momenten die Auswirkungen seiner Handlungen bzw. Entscheidungen sofort sehen können. Für das Generative System bedeutet dies, dass es Lösungen unmittelbar, d.h. annähernd in Echtzeit, erzeugen muss. Erst diese Unmittelbarkeit ermöglicht ein Verständnis der Effekte, welche durch bestimmte Veränderungen von Regeln bzw. Kriterien hervorgerufen werden.

---

<sup>4</sup> Siehe bspw. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*: John Wiley & Sons.

## 4. Wahl der generativen Methode

### 4.1. Direkte und iterative Methoden

Ein wesentlicher Punkt für die Entwicklung eines interaktiven Generativen Systems (im Sinne eines den Entwurfsprozess unterstützenden Systems) betrifft die Wahl der Generativen Methode, da diese das Verhalten des Systems bei der Problemlösungssuche bestimmt.

Bei Generativen Systemen wird im Folgenden zwischen direkten und iterativen Verfahren unterschieden. Direkte Verfahren liefern nach endlicher Zeit eine exakte Lösung für ein Problem. Sie beruhen meist auf einer umfassenden analytischen Durchdringung des Problems, welche die notwendigen Informationen zur Berechnung einer Lösung liefert. Rechenberg (1994) nennt diese Verfahren auch problemorientierte Verfahren. Da sie keinen Rückkopplungsmechanismus beinhalten (also die einmal generierten Lösungen nicht weiterentwickeln) ist das Lösungsverfahren stark linear ausgeprägt. Es ist kein Eingriff während des Problemlösungsprozesses möglich.

Die Funktionsweise iterativer Methoden besteht darin, sich in einem Prozess, bestehend aus Generierung und Bewertung schrittweise an eine Ideallösung heranzutasten. Diese Verfahren liefern zwar oft nur näherungsweise optimale Lösungen für ein Problem, jedoch ist es möglich nach jeder Iteration Änderungen an den Regeln bzw. Kriterien, die im Problemlösungsprozess angewandt werden, vorzunehmen (siehe 3.2).

### 4.2. Menge an nötigem Problemwissen

Ein weiteres wichtiges Differenzierungskriterium entwurfsunterstützender Generativer Systeme ist die Menge an notwendigem Problemwissen, welche erforderlich ist, um zu brauchbaren Entwurfsvarianten zu gelangen. Da Entwurfsprobleme nie vollständig beschrieben werden können (siehe 3.1), ist es wichtig, dass das Entwurfssystem mit möglichst wenig Problemwissen auskommt, um auch für unstrukturierte Probleme Lösungen anbieten zu können. *„Denn hochgradiges Nichtwissen ist ja gerade das, was ein Problem zum Problem macht“* (Rechenberg, 1994, p. 218).

Wie im letzten Abschnitt 4.1 beschrieben, arbeiten direkte Verfahren ohne Rückkopplungsmechanismus. Das bedeutet, dass direkte Systeme für einen bestimmten Satz an Eingangsparameter genau eine Lösung liefern und nicht wie bei iterativen Verfahren die erzeugten Lösungen bewertet und weiter optimiert werden. Das direkte Verfahren benötigt daher eine große Menge an Wissen über das Problem, welche dazu verwendet wird, den Generativen Mechanismus so zu beschreiben, dass für alle möglichen Eingangsparameter

sinnvolle Lösungen entstehen. Iterative Methoden dagegen kommen mit wenig Problemwissen für die Generierung einer Lösung aus, benötigen dafür aber einen aufwändigen Evaluations-Mechanismus. Basierend auf den Evaluationsergebnissen werden brauchbare Lösungsvarianten in einem iterativen Prozess verbessert, bis sie eine bestimmte Qualität erreichen.

In diesem Zusammenhang ist es wichtig, sich den Unterschied zwischen Problemwissen und Evaluationskriterien zu verdeutlichen. Problemwissen umfasst die genaue Analyse eines Ist-Zustands sowie die präzise Angabe der Schritte, die auszuführen sind, um zu einem bestimmten Soll-Zustand zu gelangen. Evaluationskriterien dagegen geben an, welche Eigenschaften einer Lösungsvariante wie bewertet werden sollen. Da es in der Regel einfacher ist, Evaluationskriterien zu ergänzen, als einen generativen Mechanismus an ein neues Problem anzupassen, können iterative Verfahren, die wenig Problemwissen erfordern im Vergleich zu direkten Verfahren leichter an sich ändernde Problemstellungen angepasst werden (siehe 3.1).

Zur Kategorie von Ansätzen, die viel Problemwissen erfordern, zählen die Methoden der Logischen Programmierung (Coyne, 1988), der klassischen Shape Grammar (Duarte, 2000; Stiny & Mitchell, 1978) und der reinen Constraint-Based Systeme (Li, Frazer, & Tang, 2000; Medjdoub & Yannou, 2001). Diese Methoden eignen sich insbesondere zur Bearbeitung gut definierter Probleme, wie z.B. der Nachahmung bestimmter Formen basierend auf Formgrammatiken (Shape Grammar). Zu den Ansätze, die wenig Problemwissen verlangen, zählen die Methoden der Zelluläre Automaten (Batty & Xie, 1994; Coates, Healy, Lamb, & Voon, 1996), agentenbasierten Systeme (Coates & Schmid, 2000) und Evolutionären Algorithmen (Hower, 1997; Jo & Gero, 1998; M A Rosenman, 1997).

### **4.3. Lösungsalgorithmen**

Die automatische Lösung von Layoutproblemen durch computerbasierte Methoden ist ein zentrales Thema in der Anwendung Künstlicher Intelligenz im Bereich der Architektur. Layoutprobleme sind in der Regel sehr komplexe Probleme, welche eine Vielzahl von Anforderungen erfüllen müssen. Mit jedem Kriterium, dass bei einem Layoutentwurf berücksichtigt werden soll (z.B. Anzahl der Räume) steigt die Anzahl der Lösungsmöglichkeiten exponentiell an (March & Steadman, 1971). Aus der Perspektive der Komplexitätstheorie fallen Layoutprobleme in die Kategorie der sogenannten NP-vollständigen Probleme. Zur computerbasierten Lösung solcher Probleme wurden seit Anfang der 60er Jahre (Whitehead & Eldars, 1964) verschiedene Methoden entwickelt.

Zur Bearbeitung von Problemen, bei denen die Lösung noch nicht in den Vorgaben enthalten sein soll, eignen sich insbesondere Evolutionäre Strategien (ES). Die an der biologischen Evolution orientierten Methoden lassen in einem zirkulären Trial-and-Error Prozess Lösungen entstehen, welche mittels zufälliger Variationen schrittweise an bestimmte Anforderungen (Evaluations- oder Fitnesskriterien) angepasst werden. Dabei ist es nicht nötig konkrete Verbesserungsanweisungen (wie sie im Falle direkter Methoden formuliert werden müssen) zu definieren (Rechenberg, 1994). Entscheidend für diesen Prozess sind die Evaluationskriterien. Diese geben die Richtung vor, in welcher im Lösungsraum (welcher durch die Generierungsregeln vorgegeben ist) gesucht wird<sup>5</sup>. Um die Evaluationskriterien bzw. die Generierungsregeln zu bestimmen ist eine Problembeschreibung notwendig. Diese folgt im nächsten Abschnitt.

## 5. Formalisierung des Entwurfsproblems

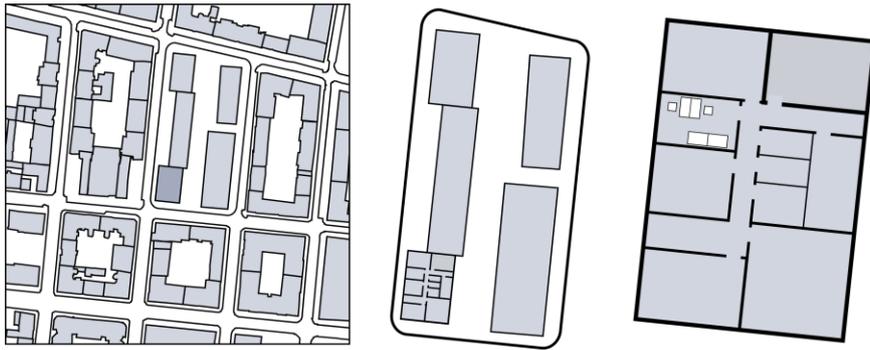
Ein wesentlicher Punkt für die Implementierung eines interaktiven Generativen Systems ist die Formalisierung des Entwurfsproblems. Durch diesen Schritt wird zugleich der Einsatzbereich des Entwurfssystems festgelegt, sprich: Was kann man mit dem System entwerfen? Je eingeschränkter dieser Bereich ist, desto eingeschränkter ist der Entwerfende bei der Suche nach Lösungen für seine Probleme, da diese wie in Punkt 3.2 dargestellt, sich oft erst während des eigentlichen Entwurfsprozesses ergeben.

### 5.1. Anordnung und Hierarchische Gliederung von Elementen

Ganz allgemein beschreibt Layout in Architektur und Städtebau die sinnfällige Anordnung verschiedener Elemente wie Parzellen, Gebäude, Räume, Zonen, Bauteile, Möbel etc. auf unterschiedlichen Maßstabsebenen (siehe Abb. 2). Maßstabsebenen bezeichnen dabei den Kontext in dem die Elemente angeordnet werden. Dieser Kontext kann für die Elemente innerhalb eines Layoutproblems unterschiedlich sein. So wird beispielsweise die Form eines Gebäudes im Kontext der städtebaulichen Nachbarschaft angeordnet bzw. verändert. Zimmer dagegen können im Kontext der durch die Form des Gebäudes begrenzenden Fläche angeordnet werden, aber auch im Kontext eines bestimmten Gebäudebereichs, wie z.B. einer Wohnung. Ein erster Schritt zur Definition von Layout wäre demnach das anordnen und skalieren von Elementen innerhalb bestimmter Kontexte. Kontexte können folglich selbst Elemente sein, innerhalb derer untergeordnete Elemente angeordnet sind.

---

<sup>5</sup> Eine ausführliche Beschreibung zur Evolutionären Strategie findet sich bei Rechenberg (1994) bzw. zu evolutionären Algorithmen bei Bentley, P. J., & Corne, D. W. (2002). An Introduction to Creative Evolutionary Systems. In P. J. Bentley & D. W. Corne (Eds.), *Creative Evolutionary Systems* (pp. 1-76). San Francisco: Morgan Kaufmann.



**Abb. 2:** Verschiedene Maßstabsebenen eines Layouts

Das Verschachteln von Elementen wird auch als hierarchische Gliederung bezeichnet. Die Art und Weise einer hierarchischen Gliederung ist situationsabhängig und muss vom Entwerfenden frei bestimmt werden können. So ist es möglich, ein Haus entweder wie oben beschrieben in einzelne Wohnungen, oder in bestimmte Zonen (wie z.B. privat und öffentlich) zu gliedern oder auf eine derartige Gliederung zu verzichten. Ferner sind auch gemischte Hierarchien denkbar. Beispielsweise könnte die Erschließung, Installationen und Abstellräume eines Gebäudes in einer höheren Hierarchieebene (Gebäudekern) untergebracht werden, während die Zimmer der einzelnen Wohnungen dieses Gebäudes auf der gleichen Ebene wie der Gebäudekern organisiert werden.

## 5.2. Relationen zwischen den Elementen

Beim Layout von Elementen müssen im vorliegenden Kontext auch funktionale Anforderungen berücksichtigt werden. Diese können beispielsweise die Beziehungen zwischen den Elementen eines Layouts umfassen, anhand derer festgelegt werden kann, welche Elemente zueinander benachbart sein sollen. Jo und Gero beschreiben Layout als *"the assignment of discrete space elements to their corresponding locations while the space elements have relationships among each other."* (Jo & Gero, 1998). Welche Elemente jedoch in Beziehung zueinander stehen können, kann auf verschiedene Weise definiert werden, da das „In-Beziehung-Setzen“ von Elementen situationsabhängig stattfindet. So ist es einerseits möglich, zwei identische Elemente miteinander in Beziehung zu setzen (Benachbarung zweier Zimmer). Andererseits ist es auch möglich, zwei (in ihrer Bedeutung) völlig unterschiedliche Elemente zu verknüpfen, wodurch z.B. die Nähe eines Raumes zu einem anderen Gebäude definiert werden kann.

Bezogen auf die Hierarchisierung von Layoutproblemen (siehe 5.1) ist es wichtig darauf hinzuweisen, dass Relationen auch über Hierarchiegrenzen hinweg definiert werden können. Dies ist erforderlich, da die Hierarchisierung zwar vorgibt, dass sich bestimmte Ele-

mente nur innerhalb eines anderen Elementes anordnen können, jedoch können funktionale Anforderungen an Elemente auch abhängig sein von Elementen, welche sich außerhalb der Grenzen des übergeordneten Elements befinden. Eine ausführliche Beschreibung zu solchen grenzübergreifenden Verknüpfungen findet sich in König und Schneider (submitted).

### 5.3. Methode zur Lösung von Layoutproblemen

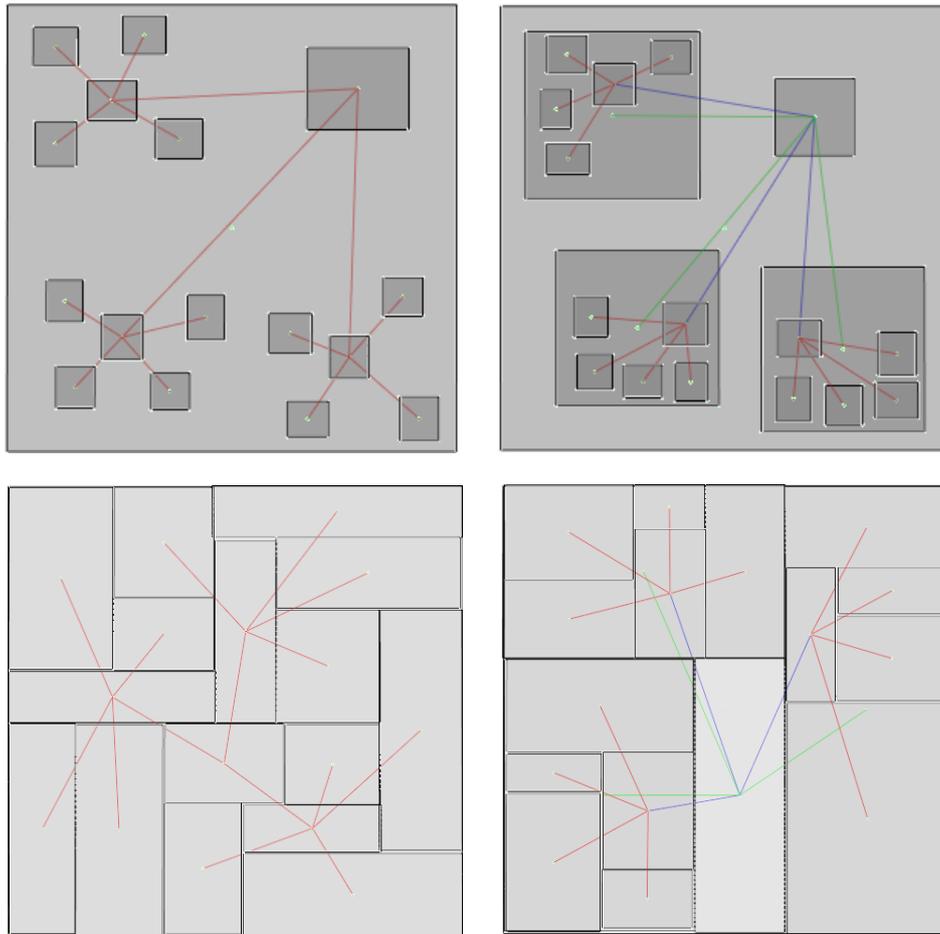
Zur Lösung von Layoutproblemen, basierend auf der hier verwendeten Definition wurde ein Generativer Mechanismus implementiert. Hierzu wurde ein iterativer Ansatz, bestehend aus einem generativen Mechanismus und einem Evaluationsmechanismus, gewählt. Der generative Mechanismus beinhaltet bestimmte Regeln zu Erstellung von Lösungsvarianten. Diese Regeln sind relativ simpel und bestehen aus der zufälligen Platzierung von Elementen innerhalb einer gegebenen Fläche. Bei den Elementen handelt es sich um Rechtecke<sup>6</sup>. Die Evaluation der generierten Lösungen basiert auf zwei Annahmen, die aus Sicht der Autoren wichtig sind, um zu sinnvollen Lösungen zu gelangen. Dies sind zum einen die überlappungsfreie und lückenlose Anordnung der Elemente, zum anderen die Berücksichtigung der unter 5.2 beschriebenen Nachbarschaftsverhältnisse einzelner Räume. Ersteres wird erreicht durch die Minimierung der Überlappungsflächen einer Anordnung, wobei die Form der Elemente (Länge und Breite) innerhalb gewisser Grenzen variabel ist<sup>7</sup>. Letzteres wird dadurch sichergestellt, dass zwei Räume, die benachbart miteinander sein sollen, keinen Abstand zueinander aufweisen und sich an je einer Kante mit einer bestimmten Länge berühren. Im Falle eines Grundrisses soll diese Anforderung den Durchgang zwischen zwei Räumen gewährleisten.

Eine detaillierte Beschreibung der verwendeten Algorithmen findet sich in König (2011). Wie topologische Relationen in hierarchisch gegliederten Layouts behandelt werden müssen wird in König und Schneider (submitted) beschrieben. Wichtig für die folgenden Überlegungen ist, dass mit Hilfe des Generativen Mechanismus Layouts erzeugt werden, deren Elemente überlappungsfrei in einer gegebenen Fläche liegen und definierte Nachbarschaftsbeziehungen erfüllen (siehe Abb. 3).

---

<sup>6</sup> Die Verwendung freier Formen würde den Such- und Lösungsraum erheblich vergrößern und damit die Berechnungsgeschwindigkeit deutlich reduzieren. Die angestrebte Echtzeitinteraktion wäre damit nicht möglich. Dennoch ist es ein Ziel zukünftiger Arbeiten, das Layoutsystem für freie Formen zu erweitern (Schneider & Koenig, forthcoming).

<sup>7</sup> Das Einführen gewisser Grenzen, bzw. fixer Proportionen ist möglich, jedoch kann dies bei zu starker Einschränkung das Finden von Lösungen verhindern.



**Abb. 3:** Lösung von hierarchisch gegliederten (rechts) und nicht hierarchisch gegliederten (links) Layoutproblemen (Oben: Definition des Problems, Unten: die dazugehörige Lösung)

Die Erzeugung einer Lösung findet in relativ hoher Geschwindigkeit statt, wodurch für den Nutzer der Eindruck eines Echtzeitfeedbacks entsteht<sup>8</sup>. Analysen zur Berechnungsgeschwindigkeit des Systems finden sich ebenfalls bei König (2011).

## 6. Darstellung und Nutzereingabe

Vergewissern wir uns nochmal des Zusammenspiels zwischen Entwerfer und Werkzeug (Punkt 3), so ist es wichtig, dass der Entwerfer das, was er mit Hilfe der Werkzeuge geschaffen hat, adäquat bewerten, bzw. wenn nötig weiterentwickeln kann. Im Falle eines Generativen Entwurfssystems ist hierfür eine geeignete Schnittstelle (Interface) zu entwickeln. Die Überlegungen, die bei der Entwicklung des im Projekt implementierten Interfaces eine besondere Rolle gespielt haben, werden im Folgenden dargestellt.

<sup>8</sup> Bei einer großen Menge an Elementen bzw. Relationen kann die Berechnung einer Lösung länger dauern und mehrere Sekunden in Anspruch nehmen. Dies hängt jedoch auch stark von der Rechenleistung der verwendeten Hardware ab.

## 6.1. Graphisches Nutzerinterface

Die Interaktion eines Nutzers mit dem Entwurfssystem findet über ein graphisches Nutzerinterface statt. Dieses Nutzerinterface ermöglicht es dem Nutzer, generierte Lösungen möglichst adäquat bewerten zu können. Für die Darstellung der Lösungen wurde eine geeignete Repräsentationsform gewählt, welche ein schnelles Erfassen ihrer relevanten Eigenschaften ermöglicht. Dass möglichst viele Eigenschaften einer Lösungsvariante erfassbar sind, ist insbesondere deshalb von Bedeutung, da es zum einen sehr viele Kriterien gibt, nach denen sich eine Lösung bewerten lässt. Zum anderen verwenden unterschiedliche Nutzer (Entwerfer) oft unterschiedliche Kriterien bei der Bewertung (Hanna, 2011).

Da sich nicht alle Eigenschaften in einer Darstellung abbilden lassen, erscheint es sinnvoll für unterschiedliche Zwecke unterschiedliche Darstellungsmethoden zu verwenden. Es eignen sich grafische Repräsentationen der jeweiligen Geometrie (in Form von Grundrissen, Stadtplänen), diagrammatische Darstellungen (wie Graphen zur Darstellung der topologischen Beziehungen) sowie textbasierte Ausgaben für relevante Kenndaten, wie etwa die Fläche von Räumen. Zwischen diesen verschiedenen Darstellungsarten muss beliebig gewechselt werden können, wobei die Datenbasis für jede Darstellung dieselbe ist. Dies entspricht im Wesentlichen dem Multiple-View-Konzept, das Rosenman und Gero (1996) für die Entwicklung von CAD Systemen vorschlagen<sup>9</sup>. Wichtig ist hierbei, dass die verschiedenen Darstellungsmodi dem Nutzer nicht nur Informationen zeigen, sondern dass dieser auch die Möglichkeit hat, mit diesen zu interagieren, sprich die dargestellten Parameter und Objekte zu ändern. Beispielsweise könnte ein Layout in Form eines Grundrisses sowie tabellarisch in Form eines Raumprogramms angezeigt werden. Änderungen im Raumprogramm hätten dann sofortige Auswirkungen auf den Grundriss zur Folge und vice versa.

## 6.2. Direkte und Indirekte Manipulation

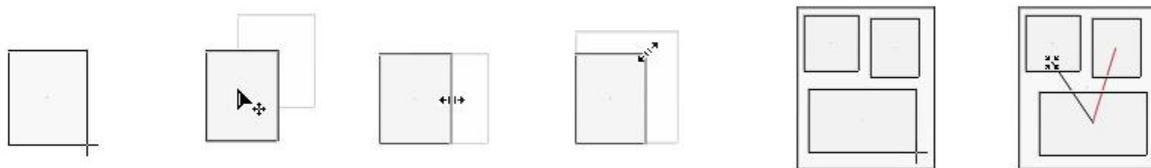
Hinsichtlich der Interaktion des Nutzers mit dem Generativen System kann zwischen direkten und indirekten Eingriffen unterschieden werden. Direkte Eingriffe bezeichnen dabei die unmittelbare Manipulation der ausgegebenen Lösung, in unserem Fall die graphische Repräsentation eines Layouts. Indirekte Eingriffe bezeichnen die Manipulation der Regeln bzw. Kriterien auf Basis derer Lösungen generiert werden.

Der Generative Mechanismus liefert beständig Lösungen für ein gestelltes Layoutproblem. Auch bei Änderungen in der Problembeschreibung reagiert das System nahezu in Echtzeit

---

<sup>9</sup> den heutigen BIM-Systemen (Building Information Modeling) liegt dasselbe Konzept zugrunde

König (2011). Um dem Nutzer die Möglichkeit zu geben, direkt in den Problemlösungsprozess einzugreifen, muss es möglich sein die Attribute der Elemente eines Layouts ebenfalls in Echtzeit ändern zu können. Dazu gehören das Verschieben und Skalieren von Elementen, das Hinzufügen und Löschen von Elementen und das Erstellen bzw. Löschen von Verknüpfungen (siehe Abb. 4). Während der direkten Interaktion mit einem oder mehreren Elementen sind die betreffenden Elemente für den Generativen Mechanismus gesperrt. Dies ist deshalb wichtig, damit der Nutzer während seines Eingriffes (z.B. Skalieren eines Elements) die Kontrolle über das Element behält.



**Abb. 4:** Verschiedene Mausinteraktionen für ein einzelnes Element

Ein weiterer wichtiger Punkt im Kontext direkter Interaktionen stellt das Festlegen von Teillösungen dar. Da der Mensch immer nur eine begrenzte Anzahl an Elementen gleichzeitig im Blick behalten kann (Miller, 1956), ist es insbesondere bei komplexeren Problembeschreibungen wichtig, dass bestimmte Teile/Gruppen von Lösungen fixiert werden können. Solche Teillösungen können einzelne oder mehrere Elemente sein, die aus bestimmten Gründen an einer bestimmten Position bleiben sollen (z.B. Größe und Position eines Funktionstraktes in einem Gebäude).

Bei der indirekten Manipulation greift der Nutzer nicht direkt in die graphische Ausgabe ein, sondern verändert die Regeln des Generativen Mechanismus oder die Bewertungskriterien des Evaluationsmechanismus. Der Nutzer steuert dadurch die Richtung in der das System den Lösungsraum durchsucht. Aufgrund der Notwendigkeit der formalen Beschreibung dieser Parameter betrifft die indirekte Manipulation die operationalen Kriterien. Die Auswirkungen von Änderungen in der Gewichtung der Bewertungskriterien, müssen möglichst sofort sichtbar sein. Er bekommt dadurch die Möglichkeit zu untersuchen, welche Auswirkungen bestimmte Bewertungskriterien (z.B. für Licht, Kompakte Form, etc.) auf das Layout haben.

Evaluationskriterien werden immer mittels numerischer Parameter definiert. Die Eingabe bzw. Änderung dieser numerischen Werte kann über Texteingaben oder sogenannte Schieberegler stattfinden (siehe Abb. 5). Für kompliziertere Evaluationskriterien (z.B. Verteilungsdichten von Elementen) könnte sich eine Definition durch Kurven oder Diagramme eignen.

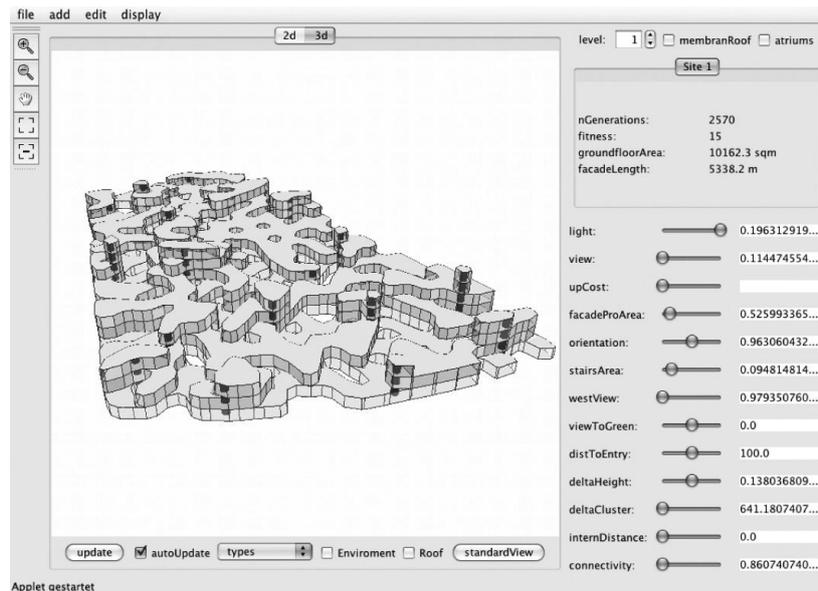


Abb. 5: Ändern der Werte der Evaluationskriterien über Schieberegler (Dillenburger, Braach, & Hovestadt, 2009)

### 6.3. Zuschreiben von Bedeutungen

Form und Funktion sind die zwei wesentlichen Kategorien, nach denen die Kriterien zur Bewertung von Architektur unterschieden werden können (Hillier, 1996). Das Aufeinanderabstimmen (oder in Einklang bringen) dieser beiden Kategorien kann als das wesentliche Ziel eines Entwurfsprozesses angesehen werden. Eine Hierarchisierung dieser Kategorien (das eine folgt dem anderen) ist dabei nicht möglich. Im Entwurfsprozess bedeutet dies einerseits, dass, um funktionale Kriterien überprüfen zu können, Annahmen über die Form gemacht werden müssen. Andererseits müssen, um Formen überprüfen zu können, Annahmen über die Funktion getroffen werden.

Übertragen wir die Unterscheidung zwischen Form und Funktion auf den Entwurf von Layouts mit der unter Punkt 5 genannten Definition, so bezeichnen die geometrischen Attribute der Elemente (Höhe, Breite, Position und deren Lage zueinander) die Form. Die Bedeutung eines Elementes, also das was ein Element repräsentiert, ist entscheiden um über die dessen Funktion bzw. Funktionieren zu urteilen. Da man beim Entwerfen, so Lawson (2005), mühelos, fast ohne es zu bemerken, zwischen den verschiedenen Bedeutungen der verwendeten Elemente wechselt, ist es wichtig, dass das Festlegen von Bedeutungen flexibel geschehen kann. Der Nutzer muss den verwendeten Elementen selbst Bedeutungen zuschreiben können, bzw. diese auch während des Problemlösungsprozesses ändern können (siehe Abb. 6).

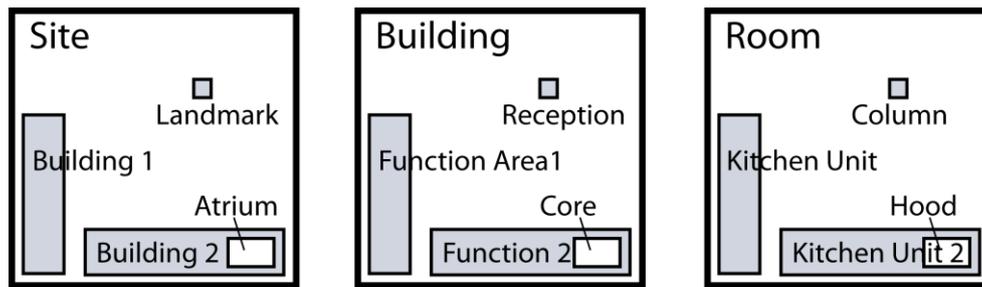


Abb. 6: Gleiche Repräsentation, unterschiedliche Interpretation

Durch die Bedeutung eines Elements können unterschiedliche Anforderungen (z.B. Mindestmaße) abgeleitet werden, welche im Generativen Mechanismus berücksichtigt werden müssen. Außerdem kann die Art und Weise der Darstellung eines Elements in Abhängigkeit seiner Bedeutung erfolgen. Wie dieses Zuschreiben von Bedeutungen bzw. Auswerten der Bedeutungen innerhalb der graphischen Nutzeroberfläche erfolgen kann, ist zum derzeitigen Projektstand noch nicht geklärt. Eine interessante Möglichkeit, die sich aus der Bedeutung der Elemente für die Repräsentation eines Layouts ergibt, ist die im Folgenden vorgestellte kontextsensitive Darstellung.

#### 6.4. Kontextsensitive Darstellung

Die Darstellung des Kontextes, erleichtert bzw. ermöglicht überhaupt erst die Bewertung eines Lösungsvorschlages, denn der Kontext, so Guski (2000), beeinflusst die Interpretation. Dies lässt sich eindrücklich an einem einfachen Beispiel (Abb. 7) zeigen: „Wenn wir z.B. einen Text lesen, werden wir eher Buchstaben erwarten als Zahlen, und wenn eine Reizkonfiguration gleichermaßen als Zahl wie als Buchstabe interpretiert werden kann, dann wird sie als Buchstabe interpretiert.“ (Guski, 2000, p. 69).

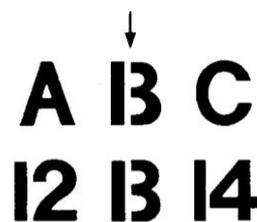
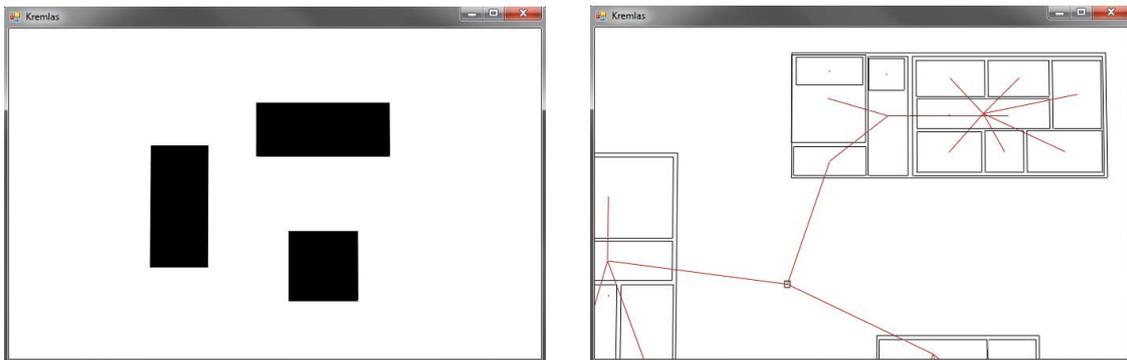


Abb. 7: Der Kontext beeinflusst die Interpretation (Guski, 2000)

Bei der Generierung eines Layouts für ein einzelnes Gebäude ist es beispielsweise notwendig das städtebauliche Umfeld, in dem sich dieses Haus befindet, darzustellen. Dies ist zum einen wichtig, um die Form (den Umriss) des Gebäudes besser beurteilen zu können, zum anderen ermöglicht es Aussagen über die innere Organisation des Gebäudes, wie z.B. die Platzierung des Haupteinganges oder die Orientierung lärmempfindlicher Räume.

Für die Repräsentation von Layoutlösungen im vorgestellten Layout-Entwurfssystem ist eine kontextsensitive Darstellung sinnvoll. Diese kann in Abhängigkeit vom Kontext und der Bedeutung der Elemente die Darstellung des Layouts anpassen. Ein Beispiel hierfür ist die bisher im System implementierte Methode zur Anpassung der Detaillierungstiefe in Abhängigkeit von der Betrachterhöhe<sup>10</sup> (siehe Abb. 8). Dies entspricht im dem Wechsel des Kontextes von städtebaulichem Umfeld zu innerer Gebäudeorganisation.



**Abb. 8:** Darstellung eines Layouts in Abhängigkeit vom betrachteten Bildausschnitt

Was bei der kontextsensitiven Darstellung als problematisch angesehen werden kann, ist, dass das, was in einem bestimmten Kontext betrachtet wird, situationsabhängig wechseln kann. Deutlich wird das an dem bereits oben genannten Beispiel der Generierung eines Hauses in einem städtischen Kontext. Hier kann einerseits die Form des Gebäudes, andererseits der Grundriss des Gebäudes dem Kontext „städtebauliches Umfeld“ gegenübergestellt werden. Auch kann der Grundriss im Kontext der Gebäudeform betrachtet werden, bzw. ein einzelnes Zimmer im Kontext des Gesamtgrundrisses. Eine Reihenfolge für das Wechseln der Kontexte ist a priori nicht festlegbar. Die Entscheidung, was wann in welchem Kontext stehen soll, muss vom Nutzer selbst entschieden werden können. Eine vollständig automatisierte Kontextsensitivität scheint nicht möglich zu sein.

## 7. Zusammenfassung und Ausblick

Generative Systeme bieten ein großes Potential zur Unterstützung von Entwurfsprozessen. Dabei muss jedoch beachtet werden, dass eine algorithmische Lösung von Problemen nur möglich ist, wenn diese gut definiert, also operationalisierbar sind. Entwurfsprobleme sind in aller Regel schlecht-definierte Probleme, sie entziehen sich einer vollständigen formalen Beschreibung. Die Lösung solcher Probleme ist abhängig von subjektiven und kontextabhängigen Faktoren. Um Generative Systeme optimal zur Unterstützung von Entwurfsprozessen einsetzen zu können, ist die Einbeziehung menschlicher Fähigkeiten in den computer-

<sup>10</sup> Auch bekannt als semantischer Zoom

basierten Problemlösungsprozess notwendig. Im vorliegenden Arbeitspapier wurden Anforderungen für ein interaktives Generatives System formuliert. Es wurden Methoden vorgestellt, wie diese Anforderungen im Kontext eines Systems zur Lösung von Layoutproblemen berücksichtigt werden können. Diese Methoden wurden in Softwaremustern umgesetzt, um deren Funktionalität überprüfen zu können.

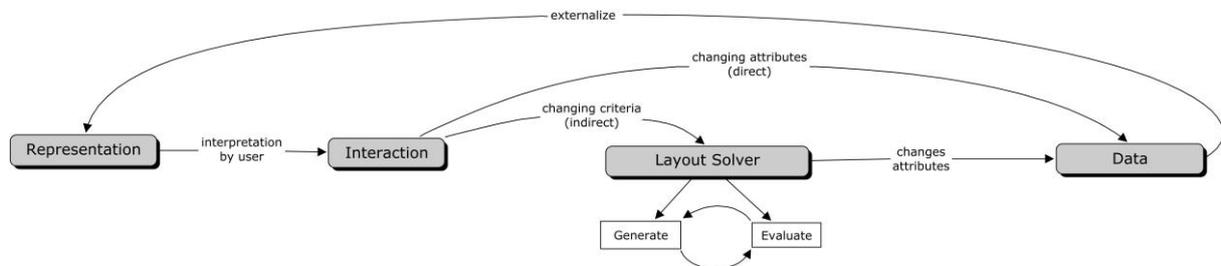


Abb. 9: Teilaspekte des Entwurfssystem

Das bisher entwickelte System kann in vier Teilaspekte gegliedert werden. Diese sind in Abb. 9 dargestellt und beinhalten die graphische Repräsentation, die Interaktion mit eben dieser, den generativen Mechanismus und die Organisation der Daten. Zwischen diesen Teilaspekten besteht ein enger Zusammenhang. Im Zentrum steht der Layout Solver, welcher Lösungen erzeugt, die hinsichtlich der Bewertungskriterien (Dichte Packung und topologische Beziehungen) optimiert sind. Dabei hat der Nutzer jederzeit die Möglichkeit, auf Grundlage der durch das graphische Interface dargestellten Lösungsvariante in den Problemlösungsprozess einzugreifen. Dies geschieht durch Ändern der Problembeschreibung, wobei durch direkte Interaktion geometrische Eigenschaften der Elemente verändert, durch indirekte Interaktion Kriterien für die Generierung verändert werden. Durch das unmittelbare Feedback entsteht ein Entwurfskreislauf, innerhalb dessen der Entwerfer einen bestimmten Raum an möglichen Lösungen flexibel durchsuchen und bewerten kann.

Für das Thema der Interaktion wurden hauptsächlich Fragestellungen behandelt, die sich mit dem Einbinden eines Entwurfssystems in den menschlichen Problemlösungsprozess befassen. Fragen der Ergonomie (also der optimierten Nutzereingabe) wurden nicht behandelt. So ist für die Weiterentwicklung des vorgestellten Systems die Anbindung an ein haptisches Interface angedacht (Knecht, 2011).

## Referenzen

- Batty, M., & Xie, Y. (1994). From cells to cities. *Environment and Planning B: Planning and Design* 21(7), 31-48.
- Bentley, P. J., & Corne, D. W. (2002). An Introduction to Creative Evolutionary Systems. In P. J. Bentley & D. W. Corne (Eds.), *Creative Evolutionary Systems* (pp. 1-76). San Francisco: Morgan Kaufmann.
- Coates, P., Healy, N., Lamb, C., & Voon, W. L. (1996). The use of Cellular Automata to explore bottom up architectonic rules. In J. Rossignac & F. Sillion (Eds.), *Eurographics '96*. Poitiers: Blackwell Publishers.

- Coates, P., & Schmid, C. (2000). Agent Based modelling. from CECA - The centre for computing & environment in architecture, University of East London School of Architecture: <http://uelceca.net/research/ECAADE/agentnotes%20the%20paper398liverpool%201999.pdf>
- Coyne, R. (1988). *Logic Models of Design*. London: Pitman Publishing.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.
- Dillenburger, B., Braach, M., & Hovestadt, L. (2009). *Building design as individual compromise between qualities and costs: A general approach for automated building generation under permanent cost and quality control*. Paper presented at the Joining Languages, Cultures and Visions: CAADFutures 2009.
- Duarte, J. P. (2000). *Customizing mass housing: a discursive grammar for Siza's Malagueira houses*. Massachusetts Institute of Technology.
- Eckert, C., Kelly, I., & Stacey, M. (1999). Interactive generative systems for conceptual design: An empirical perspective. *Artif. Intell. Eng. Des. Anal. Manuf.*, 13(4), 303-320.
- Ernst, G., & Newell, A. (1969). *GPS: A Case Study in Generality and Problem Solving*. New York: Academic Press.
- Franck, G., & Elezkurtaj, T. (2002). Design Methods. Retrieved 10.08.2008, from [http://www.iemar.tuwien.ac.at/assets/docs/design\\_methods.pdf](http://www.iemar.tuwien.ac.at/assets/docs/design_methods.pdf)
- Gänshirt, C. (2007). *Tools for Ideas: An Introduction to Architectural Design*. Basel: Birkhäuser
- Getzels, J. W., & Csikszentmihalyi, M. (1967). Scientific creativity. *Science Journal*, 3, 80-84.
- Guski, R. (2000). *Wahrnehmung: eine Einführung in die Psychologie der menschlichen Informationsaufnahme*: Kohlhammer.
- Hanna, S. (2011). *Design agents and the need for high-dimensional perception*. Paper presented at the Design Computing and Cognition '10, Stuttgart.
- Hillier, B. (1996). *Space is the machine: a configurational theory of architecture*: Cambridge University Press.
- Hower, W. (1997). Placing computations by adaptive procedures. *Artificial Intelligence in Engineering* 11, 307-317.
- Jo, J. H., & Gero, J. S. (1998). Space layout planning using an evolutionary approach. *Artificial Intelligence in Engineering*, 12, 149-162.
- Knecht, K. (2011). Augmented Urban Model: Ein Tangible User Interface zur Unterstützung von Stadtplanungsprozessen.
- König, R. (2011). Generierung von Grundriss-Layouts mittels hybrider Evolutions-Strategie.
- König, R., & Schneider, S. (submitted). Hierarchical structuring of layout problems in an interactive evolutionary layout system. *AIEDAM Special Issue DCC10*.
- Lawson, B. (2005). Oracles, draughtsman and agents: the nature of knowledge and creativity in design and the role of IT. *Automation in Construction*, 14(3), 383-391.
- Lawson, B. (2006). *How Designers Think: The Design Process Demystified* (4 ed.). Oxford: Architectural Press.
- Li, S.-P., Frazer, J. H., & Tang, M.-X. (2000). *A Constraint Based Generative System for Floor Layouts*. Paper presented at the Fifth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA), Singapore
- March, L., & Steadman, P. (1971). *The Geometry of Environment: An Introduction to Spatial Organization in Design*. London: R.I.B.A. Press.
- Medjdoub, B., & Yannou, B. (2001). Dynamic space ordering at a topological level in space planning. *Artificial Intelligence in Engineering*, 15, 47-60.
- Miller, G. A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, 63, 81-97.
- Negroponte, N. (1970). *The architecture machine*: M.I.T. Press.
- Rechenberg, I. (1994). *Evolutionsstrategie '94*: Frommann Holzboog.
- Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*, 4, 155-169.
- Röpke, J. (1977). *Die Strategie der Innovation: Eine systemtheoretische Untersuchung der Interaktion von Individuum Organisation und Markt im Neuerungsprozess*. Tübingen: Mohr Siebeck.
- Rosenman, M. A. (1997). The generation of form using an evolutionary approach. In D. Dasgupta & Z. Michalewicz (Eds.), *Evolutionary Algorithms in Engineering Applications* Springer.
- Rosenman, M. A., & Gero, J. S. (1996). Modelling multiple views of design objects in a collaborative CAD environment. *Computer-Aided Design*, 28(3), 193-205.
- Schön, D. A. (1983). *The reflective practitioner: how professionals think in action*: Basic Books.
- Schön, D. A. (1992). Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems*, 5(1), 3-14.
- Simon, H. (1969). *The Science of the Artificial*: MIT Press.
- Stiny, G., & Mitchell, W. J. (1978). The Palladian Grammar. *Environment and Planning B: Planning and Design*, 5(1), 5-18.