



**DI Martin Emmerer**

**Architektur Routine(n)**

Maschinelle Evaluierung architektonischer Entwurfslösungen  
auf Grundlage einer formalisierten Gebäudespezifikation.

**Dissertation**

zur Erlangung des akademischen Grades  
Doktor der technischen Wissenschaften (Dr. techn.).  
Vorgelegt an der Fakultät für Architektur der Technischen Universität Graz.

Gutachter

**Urs Hirschberg, Univ.-Prof. Dipl.-Arch. Dr.sc.ETH**

Leiter des Instituts für Architektur und Medien,  
Technische Universität Graz.

**Ludger Hovestadt, Prof. Dr.**

Stellvertretender Leiter des Instituts für Technologie in der Architektur,  
Department Architektur, ETH Zürich.

Graz, 2014

Martin Emmerer



# Architektur Routine(n)

Maschinelle Evaluierung architektonischer Entwurfslösungen  
auf Grundlage einer formalisierten Gebäudespezifikation.

#### EIDESSTATTLICHE ERKLÄRUNG | AFFIDAVIT

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/ resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present doctoral dissertation.

Datum / Date

Unterschrift / Signature

# INHALT



Kurzversion	VII
Abstract	IX
Vorwort	XI
<b>Einleitung</b>	<b>1</b>
<b>1 Raumprogramme – Architektur à la Carte?</b>	<b>18</b>
1.1 Funktionsbereiche	21
1.2 Merkmale	21
1.3 Beziehungen	24
1.4 Organigramme	25
1.5 Schlussfolgerungen	28
<b>2 Graphen – Lernen von Königsberg</b>	<b>31</b>
2.1 Das Königsberger Brückenproblem	31
2.2 Topologie und Graphentheorie	36
2.3 Liniennetze	39
2.4 Graphen in der Architektur	42
2.5 Schlussfolgerungen	47
<b>3 Formalisierte Gebäudespezifikation</b>	<b>50</b>
3.1 Datenmodellierung	50
3.2 Merkmalstypen	52
3.3 Präferenzfunktionen	57
3.4 Extensible Space Allocation Language – XSAL	60
3.5 Design Goal Manager	69
3.6 Schlussfolgerungen	71
<b>4 Building Information Model</b>	<b>73</b>
4.1 Historische Entwicklung	74
4.2 Draftpad	77
4.3 Schlussfolgerungen	87

<b>5 Algorithmische Explikation</b>	<b>88</b>
5.1 Explikation	89
5.2 Implementierung	93
5.3 Schlussfolgerungen	97
5.4 Inhaltsverzeichnis Merkmalskatalog	99
<b>6 Anwendungen</b>	<b>100</b>
6.1 Fitness einer Entwurfslösung	101
6.2 Neue Instrumente für den Architekturentwurf	113
6.3 Schlussfolgerungen	121
<b>Conclusion – Der Kreis schließt sich</b>	<b>122</b>
<b>Anhang</b>	
Merkmalskatalog	135
Literatur	169
Abbildungen	179
Curriculum Vitae	183

## KURZVERSION



### Architektur Routine(n)

#### Maschinelle Evaluierung architektonischer Entwurfslösungen auf Grundlage einer formalisierten Gebäudespezifikation.

Der architektonische Entwurfsprozess hat sich schon lange größtenteils auf den Computer verlagert. Die Beurteilung von Gebäuden ist bis heute dennoch eine ausschließlich menschliche Domäne geblieben. Eine Vielzahl von Entwurfparametern, wie sie zum Beispiel in Raumprogrammen von Wettbewerbsauslobungen beschrieben werden, könnten jedoch auch maschinell erfasst werden. Für das Arbeits- und Lernumfeld von Architektinnen und Architekten wäre damit aus mehreren Gründen ein Mehrwert verbunden.

Im Zentrum der vorliegenden Arbeit steht die Entwicklung eines offenen Systems zur maschinellen Evaluierung von Gebäudeentwürfen. Dieses Vorhaben verlangt nach einer durchgehenden und lückenlos geschlossenen Formalisierung des gesamten Kreislaufs, von der verbalen Beschreibung von Architekturqualitäten, bis hin zur Erfassung der interessierenden Eigenschaften konkreter Entwurfslösungen in einem Gebäudedatenmodell.

Zunächst wird zu diesem Zweck die EXTENSIBLE SPACE ALLOCATION LANGUAGE (XSAL) vorgestellt, der Vorschlag eines offenen Standards zur Beschreibung von Gebäudemerkmalen auf Basis der Metasprache XML. Gebäudespezifikationen in XSAL Notation können nicht nur von Menschen gelesen werden, sondern stellen die darin enthaltenen Informationen auch für eine maschinelle Datenverarbeitung bereit.

Der zweite Schwerpunkt liegt in der Entwicklung eines geeigneten Gebäudedatenmodells. Aufgrund der ständig steigenden technischen Anforderungen an Bauwerke sind die meisten Building Information Models heute zu hoch komplexen Datenbanken herangewachsen. Zur Evaluierung von Entwurfslösungen wird ein simpleres, vom Autor entwickeltes, assoziatives Gebäudedatenmodell vorgeschlagen, dessen Entitäten sich semantisch mehr am entwerferischen Denken bzw. an der Art, wie Architektur beschrieben wird, als am bautechnischen Konstruieren orientieren.

Die ARCHITEKTUR ROUTINE(n) stellen das dritte und verbindende Element zwischen Gebäudespezifikation und digitaler Modellwelt dar. Hierbei handelt es sich um Algorithmen zur Entdeckung der interessierenden Entwurfsqualitäten im Gebäudedatenmodell. Um die häufig vagen Begriffe der Alltagssprache, welche zur Beschreibung von Entwurfsaufgaben benutzt werden, in mess- und berechenbare Angaben umzuwandeln, wurde das Konzept der ALGORITHMISCHEN EXPLIKATION entwickelt. Durch die wiederholte Anwendung dieses Verfahrens entsteht ein individuell erweiterbarer Wortschatz, der nicht nur a priori an intersubjektive Vorstellungen geknüpft ist, sondern auch einen Dialog zwischen Mensch und Maschine erlaubt.

Anhand der vom Autor entwickelten Software ARCHILL.ES wird schließlich gezeigt, wie die Potenziale eines solchen expliziten Vokabulars in den Workflow der Praxis integriert und für die Anwendung generativer Entwurfsmethoden durch die Bildung von Fitnessfunktionen genutzt werden kann (s. WWW.ARCHILL.ES). Darüber hinaus wird eine neue Form eines sprachbasierten Data-Minings in einem Building Information Model eingeführt.





## ABSTRACT



### Architektur Routine(n)

#### **Machinic Evaluation Of Architectural Design Solutions Based On A Formalized Building Specification.**

For quite some time now the architectural design process has to a large extent been driven by and developed with the aid of computers. The assessment of buildings, on the other hand, has to this day remained an exclusively human domain. A variety of design parameters, which for instance are found in the space allocation programs of competition briefs, could potentially be captured and processed by a machine and thereby generate an added value for the work- and learning-environments of architects.

The main focus of this work is the development of an open system that enables the machine-driven evaluation of a building design. Such a system requires a continuous and uninterrupted formalization of the whole design cycle – from the verbal description of architectural qualities to the capturing of relevant aspects of specific design solutions presented via a Building Information Model (BIM).

First, the EXTENSIBLE SPACE ALLOCATION LANGUAGE (XSAL) is introduced: the proposal of an open standard to describe the building criteria based on the meta-language XML. Building specifications noted in XSAL can be read and understood by humans and its inherent information can also be processed by machines.

The second emphasis of this work lies in the development of a suitable building-datamodel. Due to constantly rising technical requirements for buildings, most of today's Building Information Models (BIM) have grown to become highly complex data-bases. The author proposes a much simpler associative building-datamodel for the evaluation of design solutions. Rather than drawing from civil engineering, engineering and construction, this model relates to the notion of 'design thinking' and to the way architecture is described.

The ARCHITEKTUR ROUTINE(n), a collection of algorithms to detect and explore qualities of interest or of concern, are the third and connecting element between the building specification and the digital model. The ALGORITHMIC EXPLICATION was developed in order to find a more precise language for describing a design task and to convert the often vague terms of our everyday language into quantitative (measurable) and consequently computable form. Re-iterative and repeated use of this method creates an explicit and expandable vocabulary, which is not based on the a priori inter-subjective idea but which enables a dialogue between man and machine.

Finally, ARCHILL.ES, a software developed by the author, is used to demonstrate how the potential of such an explicit vocabulary can be integrated into the work-flow of everyday practice and how, through the definition of fitness functions, it can be applied to generative design methods (s. WWW.ARCHILL.ES). Furthermore, a new form of a language based Data-Mining is introduced to Building Information Models.



DJ at Work, 2013 © photodune.net

## VORWORT



*Flinke Hände verschieben Slider und drehen an Knöpfen. Parameter werden definiert und Werte gesetzt. Zugleich generiert sich eine räumliche Struktur, die sich mit jeder Justierung einer Variablen verändert. Formen entwickeln sich im Fluss der Zeit als kontrolliert metastabile Erscheinungen. Angetrieben durch Interaktion strebt ein dreidimensionales Visual in Richtung Selbstorganisation. Das Setup steht. Der Raum bebt. Die Unité d’Habitation als typologische Baseline, darüber ein fein getunter Variablenmix: Der Haus DJ ist am Werk!*

Seit geraumer Zeit (1999) verfolgt mich eine Idee – Der *Haus DJ*. – Wer ist dieser virtuose *Entwerfer am Mischpult*, der scheinbar spielerisch seine Arbeit tut. Und welche Art von Instrument benutzt er für sein Spiel mit Raum?

Ich danke allen, die mich dabei unterstützt haben, ein schärferes Bild zu zeichnen! Urs Hirschberg, der diese Dissertation als Betreuer und Mentor begleitet hat. Und Ludger Hovestadt, dem zweiten Gutachter, der mir in unseren Gesprächen neue Sichtweisen eröffnete. Für das Lektorat danke ich Martina Bürgermeister, Angelika Langmann, Claudia Kappl, Bernhard Lechner und Markus Wallner-Novak.

Martin Emmerer, Graz 2014



# EINLEITUNG

*»Es gibt aber, außer der Anschauung,  
keine andere Art, zu erkennen, als durch Begriffe«<sup>1</sup>*

Immanuel Kant 1781

Raumprogramme beschreiben notwendige Eigenschaften und funktionale Anforderungen an zu planende Gebäude. Solche Schriftstücke waren bisher immer von Menschen an Menschen gerichtet. Dies ist auch nicht weiter verwunderlich. Die Anforderungen an zukünftige Gebäude entspringen größtenteils menschlichen Bedürfnissen, und Architektur wurden in der Geschichte der Baukunst bis heute ausschließlich von Menschen, sowohl in Auftrag gegeben als auch entworfen. Aus diesem Grund wurden auch die architektonischen Entwurfsaufgaben stets in natürlicher Sprache artikuliert. Solche von Menschen verfasste Programme lassen in der Folge auch nur eine subjektive Betrachtung und Beurteilung der Ergebnisse zu. Ein Raumprogramm sollte die Anforderungen an ein zu planendes Gebäude erkennen lassen. Da von einem noch nicht entworfenen Gebäude noch keine sinnliche Anschauung existiert, kann eine Darstellung desselben nur über Begriffe erfolgen.

Der Wunsch nach einer „gut orientierten“ Wohnung oder die Forderung, dass Raum A „nahe bei“ Raum B situiert sein soll, stellen für Menschen scheinbar eindeutige Planungsvorgaben dar. Wir verbinden mit dem Begriff „Orientierung“ genauso eine bestimmte Vorstellung, wie mit dem der „Nähe“. Bei genauerer Betrachtung sind diese Begriffe jedoch äußerst vage. Welchen Kriterien muss eine Wohnung entsprechen, die man als „gut orientiert“ bezeichnet? Bezieht sich diese Eigenschaft ausschließlich auf die Ausrichtung von Fensteröffnungen? Oder was genau bedeutet „Nähe“ in Bezug auf zwei Räume in einem Gebäudeentwurf? Ist dafür ein direkter Zugang erforderlich? Oder darf zwischen den betreffenden Räumen auch ein kurzer Erschließungsgang liegen? Wenn ja, wie lange darf der Weg maximal sein? In Ermangelung klarer Standards bei der Benennung der meisten räumlichen Phänomene kann das Urteil über viele Qualitäten eines Gebäudeentwurfs nur subjektiv gefällt werden.

Der architektonische Entwurfsprozess hat sich schon lange größtenteils auf den Computer verlagert. Beinahe jeder Entwurf liegt heute früher oder später in einer digitalen „Entwurfswelt“ vor. Die Beurteilung von Gebäuden ist in der Praxis dennoch eine ausschließlich menschliche Domäne geblieben. Bei einer kritischen Betrachtung der heutigen Arbeits- und Lernbedingungen von Architektinnen und Architekten lässt sich eine Reihe von Gründen isolieren, die dafür sprechen, dass mit einer maschinellen Evaluierung von Gebäudeentwürfen ein Mehrwert verbunden ist:

Der erste Grund ist im Bereich der klassischen Entwurfspraxis zu finden. Durch den Einsatz von CAD-Programmen und assoziativen Gebäudedatenmodellen (BIM) lassen sich architektonische Entwürfe mittlerweile zwar wesentlich komfortabler entwickeln als zuvor auf dem Reißbrett; Feedback über die Qualitäten eines Entwurfs geben diese Softwareumgebungen jedoch nicht. Jeder architektonische Entwurf stellt eine Hypothese dar. Ob und wie weit den vielfältigen Anforderungen an ein zukünftiges Gebäude nachgekommen wird, unterliegt

---

<sup>1</sup> Kant 2011, 120.

während des gesamten Entwurfsprozesses allein der Urteilskraft der Entwerferin oder des Entwerfers. Durch das Fehlen eines begleitenden Feedbacks wird Architektinnen und Architekten nichts von der Multitasking-Aufgabe Entwerfen abgenommen. Ob ein Gebäudeentwurf alle gewünschten Parameter erfüllt, ist keineswegs offensichtlich.

Eine solche Arbeitsumgebung kann nach Robin M. Hogarth im Allgemeinen als eher „boshaft“ bezeichnet werden.<sup>2</sup> Nach Hogarth existiert eine weite Bandbreite strukturell unterschiedlicher Lernumwelten, welche von „freundlich“ (*kind*) bis „boshaft“ (*wicked*) reicht. Freundliche Umwelten lassen uns die Konsequenzen unseres Verhaltens deutlich spüren, indem sie konsistentes und klares Feedback geben. Durch die Konsequenzen unserer Entscheidung lernen wir, ob und in welchem Ausmaß das eingeschlagene Verhalten zieldienlich oder zielhinderlich ist.<sup>3</sup> Im Gegensatz dazu erteilen „boshafte“ Umwelten uneindeutige, undeutliche, zu komplexe, zeitlich-verzögerte oder manchmal auch keine Rückmeldungen. In diesen Umwelten wird nach Hogarth adaptives Lernen erschwert, welches verhaltensorientiert, also erst durch die Wechselwirkung von Handlung und Erfahrung eintritt. Von etwas, das sich unserer Erkenntnis entzieht, können wir auch nichts lernen.<sup>4</sup> Bis heute können nur wenige, der in Raumprogrammen geforderten Merkmale in einem vorliegenden Gebäudeentwurf explizit festgestellt werden. Manchmal sind zwar die Prüfverfahren bekannt, jedoch sind mit diesen in vielen Fällen zeitraubende, teilweise stumpfsinnige Tätigkeiten verbunden. Keiner unserer fünf Sinne ist in der Lage, Merkmale wie Fluchtweglänge, Raumgröße oder Tageslichtquotient<sup>5</sup> direkt (ad hoc) wahrzunehmen. Für jeden zugehörigen Nachweis der Erfüllung solcher Eigenschaften sind viele Handgriffe und teilweise mühsame Hilfskonstruktionen notwendig, die mentale Anstrengung erfordern.

Ein differenziertes maschinell generiertes Feedback über den Grad der Erfüllung der im Raumprogramm formulierten Ziele könnte Architektinnen und Architekten bei der Entwicklung eines Gebäudeentwurfs auf ähnliche Weise unterstützen, wie die Rechtschreibprüfung eines Textverarbeitungsprogramms Literatinnen und Literaten bei der Erstellung eines Manuskripts. Schon der Mathematiker und Philosoph Alfred North Whitehead, der gemeinsam mit Bertrand Russell die *Principia Mathematica* verfasste, war der Überzeugung, dass die Zivilisation ihren Fortschritt dadurch erzielt, „indem sie die Zahl der wichtigen Vorgänge vermehrt, die wir ohne Nachdenken ausführen können.“<sup>6</sup> Derartige Tätigkeiten werden in der Regel als Routinen bezeichnet.<sup>7</sup> Der Ausdruck Routine bezeichnet eine Form von handlungsbezogenem Wissen, das durch einmalige oder wiederholende Erfahrungen erworben wurde. Noch besser wäre es allerdings, diese von Computern automatisiert ausführen zu lassen. Routinen eignen sich vorzüglich zu einer Implementierung, da sie per se bereits Programme sind. Durch eine maschinelle Überwachung der maßgeblichen Entwurfsziele und ein klares Feedback durch den Computer könnte der Entwerferin oder dem Entwerfer, wie es schon Douglas Engelbart im Jahre 1962 in seinem Aufsatz *Augmenting Human Intellect*<sup>8</sup> prophezeite, ein wesentlich „freundlicheres“ Lern- und Arbeitsumfeld geboten und dadurch der Umgang mit der Komplexität erleichtert werden.

---

<sup>2</sup> Vgl. Hogarth 2001.

<sup>3</sup> Vgl. Betsch 2005, 267.

<sup>4</sup> Vgl. Emmerer 2014, 59f.

<sup>5</sup> Der Tageslichtquotient ist ein Maß zur Beurteilung der Tageslichtversorgung von Räumen.

<sup>6</sup> Alfred North Whitehead, zit. n. Gigerenzer 2008, 49.

<sup>7</sup> Vgl. Betsch 2005, 262.

<sup>8</sup> Vgl. Engelbart 1962, 5.

Der zweite Grund für die maschinelle Evaluierung von Entwurfslösungen betrifft ein instrumentelles Defizit generativer Entwurfsmethoden. Seit den 1960er Jahren träumt die architektonische Computeravantgarde davon, Formfindungsprozesse auf die Maschine übertragen zu können. Die Idee, Architektur als sich selbst erschaffendes, autopoietisches System zu betrachten, in welches der Mensch nur noch von außen durch das Abstimmen beziehungsweise Mischen von Parametern eingreift, fasziniert seit den Anfängen des Strukturalismus. *Kybernetik*, *autonome Architektur* und *Selbstorganisation* sind Schlagworte, die in diesem Zusammenhang häufig benutzt werden. Die Beweggründe für solche Visionen sind vielfältig. In jedem Fall spiegelt sich darin die Hoffnung, Computer könnten uns mit atemberaubenden neuen räumlichen Komplexen überraschen – Gebäude generieren, zu deren Schaffung wir niemals selbst in der Lage wären. Für die Verwirklichung dieser Idee wäre so mancher Architekt sogar bereit, etwas von der direkten Kontrolle über die Form abzugeben.<sup>9</sup> Beispielsweise wird in diesem Feld mit dem Einsatz evolutionärer Algorithmen experimentiert. Variantenbildung und Selektion sind bei diesem oder ähnlichen Ansätzen von zentraler Bedeutung. Aus einer häufig großen Zahl von Entwurfsvarianten müssen dabei vom Computer die „fittesten“ identifiziert werden. Fitness bedeutet hierbei allgemein die Maximierung von vorab definierten Zielfunktionen. Im Speziellen, bezogen auf Gebäude handelt es sich hierbei um den Grad der Erfüllung der im Raumprogramm beschriebenen funktionalen Anforderungen. Die Leistungsfähigkeit solcher Systeme hängt weniger von den Such- oder Mutationsalgorithmen, als von einer möglichst umfassenden Spezifikation der Zielfunktion ab.<sup>10</sup>

Eine Vielzahl von Entwurfparametern könnte durch den Einsatz der Maschine überwacht werden. In der Architekturtheorie besteht schon seit einem halben Jahrhundert Interesse an parametrischen und generativen Entwurfsmethoden. Dennoch wurde bis heute noch keine<sup>11</sup> allgemeine Systematik für die automatisierte Erfassung räumlicher Qualitäten und Quantitäten entwickelt, welche in der Lage wäre, einerseits die aus bekannten Methoden wie beispielsweise *Spacesyntax*<sup>12</sup> oder Christopher Alexanders *Pattern Language*<sup>13</sup> stammenden Analyseparameter zu integrieren und zugleich eine modulare Erweiterung für problemspezifische Anforderungen und Experimente zulässt. Schon im Jahre 1977 wies der Architekt William Mitchell auf die Notwendigkeit von Grundlagenforschung hin, um die enormen Möglichkeiten des Computers ausschöpfen zu können.

*»Once the principles of encoding, storage, input, and output of architectural data are understood, the possibilities for development of programs to solve architectural problems of various types are virtually limitless.«<sup>14</sup>*

Die rapiden Entwicklungen im Bereich der Informationstechnik und die daraus resultierenden gestalterischen Möglichkeiten scheinen im Feld der Architektur „verführerisch“ den, in den Anfängen des *Computer Aided Architectural Designs* (CAAD) bemerkenswert klaren

---

<sup>9</sup> Vgl. Emmerer 2014, 56.

<sup>10</sup> Vgl. Franck 2009, 238.

<sup>11</sup> Zumindest ist dem Autor keine umfassende wissenschaftliche Arbeit zu diesem Thema bekannt.

<sup>12</sup> Hillier/Hanson 1984.

<sup>13</sup> Vgl. Alexander 1995.

<sup>14</sup> Mitchell 1977, 379.

Blick auf basale Themen der Parametrik ein wenig verschleiert zu haben. Gerade auf elementarer Ebene liegt hier noch großes Potential. Im Zentrum der vorliegenden Arbeit steht die Entwicklung eines offenen und individuell adaptierbaren Systems zur maschinellen Evaluierung von Gebäudeentwürfen. Dieses Vorhaben verlangt nach einer durchgehenden und lückenlos geschlossenen Formalisierung des gesamten Kreislaufs, von der verbalen Beschreibung von Architekturqualitäten, bis hin zur Erfassung der interessierenden Eigenschaften in konkreten Entwurfslösungen. Die Entwicklung eines allgemeinen formalisierten Verfahrens, welches die maschinelle Beurteilung von Gebäudeentwürfen ermöglicht, wäre nicht nur für die alltägliche Praxis des menschlichen Entwerfers eine Bereicherung, sondern würde zugleich auch die Anwendung generativer Entwurfsmethoden unterstützen. Können die in heutigen Raumprogrammen beschriebenen funktionalen Anforderungen an Gebäude erst einmal maschinell überwacht und die Abweichungen von formulierten Zielen errechnet werden, so kann auch die mathematische Aggregation dieser Werte zu einem integralen Benchmark, einer sogenannten *Fitnessfunktion* erfolgen.

## Von der Vor-Schrift zum Programm

Sollen Computer Gebäudeentwürfe evaluieren, so müssen die Entwurfsziele in maschinengerechter Form kommuniziert werden. In der Folge müssen sich die im Entwurfsprogramm enthaltenen Vorschriften auch an Maschinen richten. Für diesen Zweck ist zunächst die Vereinbarung einer gemeinsamen Sprache notwendig. Schon der Architekt und Mathematiker Christopher Alexander bemerkte, dass die natürliche Sprache nicht besonders gut dafür geeignet ist, die Struktur von Planungsproblemen darzustellen. Aus diesem Grund benutzte er für seine *Pattern Language* anstatt Worte ein System aus grafischen Symbolen. In seinem Buch *Notes on the Synthesis of Form* führt er als Beispiel hierfür den Begriff der *Sicherheit* an. Die Anforderung *Sicherheit* zu gewährleisten, besteht in unterschiedlichsten Designaufgaben, beim Entwurf eines Teekessels wie auch bei der Konzeption eines Autobahnkreuzes. Das Wort *Sicherheit* hat in diesen beiden Entwurfsaufgaben unterschiedliche Bedeutungen.<sup>15</sup>

Menschen sind im intuitiven Umgang mit vagen oder mehrdeutigen Beschreibungen geübt. Computer dagegen benötigen für jede Datenverarbeitung eindeutige und normierte Informationen. Um die bis heute nur von Menschen interpretierbaren Vorschriften in eine von Maschinen lesbare Form zu bringen, ist es notwendig, bei der Erstellung von Gebäudespezifikationen Standards einzuführen und sich dabei einer *formalen Sprache*<sup>16</sup> zu bedienen. Den syntaktischen Rahmen für dieses Unterfangen kann die generische Auszeichnungssprache XML bilden. Nicht umsonst wird XML häufig als *Lingua Franca*<sup>17</sup> der Informationstechnologie bezeichnet. XML kann auch als Dokumentenformat für Gebäudespezifikationen benutzt werden, welches neben den sprachlich formulierten Zielvorstellungen an ein

---

<sup>15</sup> Vgl. Alexander 1964, 67.

<sup>16</sup> Eine formale Sprache ist eine abstrakte Sprache, bei der im Unterschied zu natürlichen Sprachen nicht die Kommunikation, sondern eine mathematische Verwendung im Vordergrund steht. Formale Sprachen, wie beispielsweise Programmiersprachen, sind Zeichenketten, welche sich für mathematisch präzise Beschreibungen eignen. Vgl. Wikipedia dt.: Formale Sprache, 22.10.2014.

<sup>17</sup> Venezianische Kaufleute benutzten bis ins 19. Jahrhundert zur Vermeidung von sprachlichen Missverständnissen beim internationalen Handel eine künstliche Verkehrssprache. Die „Lingua Franca“ bestand aus einem mit arabischen Elementen vermischten Italienisch; so war es möglich über die Grenzen der verschiedenen Sprachgemeinschaften hinweg zu kommunizieren.



zukünftiges Gebäude auch beziehungsweise Anweisungen darüber enthält, wie die enthaltenen Daten maschinell zu verarbeiten sind. Sollen architektonische Qualitäten von Computern erfasst werden, genügt es nicht mehr, eine ungefähre Vorstellung von der Bedeutung der verwendeten Begriffe zu besitzen. Dem Computer muss eine exakte Handlungsanleitung in Form eines Algorithmus bereitgestellt werden, der beschreibt, wie jedes der durch die Begriffe bezeichneten Merkmale erhoben werden kann. Dabei ist eine technische Verbindung zwischen den zur Beschreibung bestimmter Gebäudemerkmale benutzten Begriffen und den damit assoziierten Vorstellungen herzustellen.

Beim Erlernen der (Mutter)Sprache kommt es zu einer assoziativen Verknüpfung zwischen Sinneswahrnehmungen und Begriffen. Von Kindheit an stellen wir immer wieder die Frage: „Wie heißt das?“, worauf dann eine Benennung erfolgt. Auf diese Weise gelangen wir in den Besitz von Begriffen und können in der Folge auch von Gegenständen sprechen oder schreiben, die wir nicht direkt vor Augen haben – so auch von den Eigenschaften eines Gebäudes, das es noch gar nicht gibt. Der Frage, in welcher Verbindung die Begriffe mit den benannten Gegenständen stehen, ging schon Ludwig Wittgenstein in der Mitte des 20. Jahrhunderts in seinen *Philosophischen Untersuchungen* nach.

*»Was ist die Beziehung zwischen Namen und Benanntem? – Nun, was ist sie? [...] Diese Beziehung kann, unter vielem andern, auch darin bestehen, daß das Hören des Namens uns das Bild des Benannten vor die Seele ruft, und sie besteht unter anderem auch darin, daß der Name auf das Benannte geschrieben ist, oder daß er beim Zeigen auf das Benannte ausgesprochen wird.«<sup>18</sup>*



Abb 0.1 Saussure um 1910: Das Sprachliche Zeichen

Der Schweizer Sprachwissenschaftler Ferdinand de Saussure hat bereits in der ersten Hälfte des 20. Jahrhunderts in seinen *Grundfragen der allgemeinen Sprachwissenschaften* darauf hingewiesen, dass die sprachliche Einheit „etwas Doppelseitiges“<sup>19</sup> ist. Das sprachliche Zeichen entspringt nach Saussure der Vereinigung von einer Vorstellung und einem Lautbild (Abb 0.1). Das *Bezeichnete* und die *Bezeichnung* sind in unserem Gehirn durch das Band der Assoziation verknüpft. Diese Beziehung ist laut Saussure durch Willkürlichkeit (Arbitrarität) gekennzeichnet. Die Verbindung beruht auf zwischenmenschlicher Konvention und Vereinbarung und nicht auf einer naturgegebenen Gesetzmäßigkeit.<sup>20</sup>

<sup>18</sup> Wittgenstein [erstveröffentlicht 1953] 2003, 36.

<sup>19</sup> Vgl. Saussure 2001, 77.

<sup>20</sup> Die Vorstellung eines Baums ist durch keinerlei innere Beziehung mit der Lautfolge Baum als Bezeichnung verbunden. Für Saussure beweist dies schon das Vorhandensein verschiedener Sprachen. Dieselbe Vorstellung kann in verschiedenen Sprachen unterschiedliche Bezeichnungen haben. Der Ausdruck „arbor“ kann jederzeit

Von unserem Sinnesapparat werden komplexe Reizmuster aufgenommen. Treten ähnliche Muster häufig auf, bilden diese Kategorien. Für unterschiedliche Kategorien werden Name vereinbart. Benannte Ähnlichkeitsstrukturen beziehungsweise Kategorien bilden unseren sprachlichen Wortschatz. Für eine maschinelle Beurteilung von Gebäudeentwürfen muss dem geheimnisvollen *Band der Assoziation*<sup>21</sup>, welches es uns nicht nur ermöglicht, Qualitäten eines Gebäudeentwurfs graduell zu erkennen, sondern diese auch in Begriffe zu fassen und durch Wort oder Schrift zu transportieren, ein maschinelles Pendant gegenübergestellt werden.

## Calculus!

Schon der Mathematiker und Philosoph Gottfried Wilhelm Leibniz sah das Hauptübel der natürlichen Sprache in der Mehrdeutigkeit ihres Vokabulars. Er war der Meinung, dass wissenschaftliche oder philosophische Streitigkeiten fast ausnahmslos Wortstreitigkeiten sind, das heißt aus der Unexaktheit der Sprache resultieren.<sup>22</sup> Leibniz war der Überzeugung, dass alle Erkenntnisse auch auf mathematischem Weg gewonnen werden können. In seiner Dissertation *De Arte combinatoria* schlägt er im Jahre 1666 für wissenschaftliche Argumentationen eine künstliche Sprache, die *Characteristica Universalis* vor. Der Wortschatz dieser *Plansprache* sollte aus einer überschaubaren Anzahl exakt definierter elementarer Begriffe bestehen, durch deren Kombination zusammengesetzte Begriffe gebildet werden können. Als sprechbares Zeichensystem sollte die *Characteristica Universalis* sowohl geschrieben als auch gesprochen werden. Zwischen den einzelnen Symbolen, welche für diese künstliche Sprache gewählt wurden, und den assoziierten Vorstellungen, sollte eine umkehrbar eindeutige Zuordnung – also Isomorphie<sup>23</sup> bestehen. Leibniz ging es um eine spezifische Zuordnung von Zeichen und Bezeichnetem. Die natürliche Sprache erschien dem Universalgelehrten zwar ausreichend für den alltäglichen Gebrauch, nicht aber präzise genug für eine wissenschaftliche Verwendung. Zu diesem Zweck wurde eine Sammlung von Definitionen sprachlicher Ausdrücke erstellt. In diesem „*Wörterbuch der Ideen*“<sup>24</sup> sollte die Bedeutung jedes elementaren Begriffs a priori festgelegt werden.<sup>25</sup>

*„Ganz im Geiste Descartes ging es Leibniz in seinem Werk um ein Schriftsystem, das auf einer vollständigen Analyse und Klassifikation des damaligen Wissens beruhte. Es sollte also einen Weltkatalog darstellen, dessen kleinste Elemente nicht weiter reduzierbare Begriffe wären.“<sup>26</sup>*

---

durch „Baum“ ersetzt werden. Im Gegensatz dazu sind die Bezeichnungen bestimmter Vorstellungen innerhalb einer Sprachgemeinschaft nicht beliebig. Eine von einer bestimmten Sprache gewählte Bezeichnung kann nicht einfach von der Masse der Sprachgenossen, schon gar nicht von einem einzelnen Individuum durch eine andere ersetzt werden. Vgl. Saussure 2001, 76 f.

<sup>21</sup> Vgl. Saussure 2001, 77.

<sup>22</sup> Vgl. Giesecke 1988, 120.

<sup>23</sup> Isomorphismus bezeichnet die Abbildung zwischen zwei (mathematischen) Strukturen, durch die Teile einer Struktur auf bedeutungsgleiche Teile einer anderen Struktur umkehrbar eindeutig (bijektiv) abgebildet werden. Vgl. Wikipedia, dt.: 22.1.2014.

<sup>24</sup> Blanke 1985, 140.

<sup>25</sup> Arndt 2011, 4.

<sup>26</sup> Blanke 1996, 29ff.

In der künstlichen Sprache von Leibniz wird jedem Elementarbereich eine Primzahl<sup>27</sup> zugeordnet. Komplexe, aus basalen Begriffen zusammengesetzte Begriffe werden durch das Produkt aus deren numerischen Werten berechnet. Wird dem Begriff *Lebewesen* beispielsweise die Zahl 2 und dem Begriff *vernunftbegabt* die Zahl 3 zugewiesen, so errechnet sich der zusammengesetzte Begriff *vernunftbegabtes Lebewesen* durch die Multiplikation der Zahlen 2 und 3. Dies ergibt die Zahl 6.<sup>28</sup>

In erster Linie ging es Leibniz dabei um eine Erleichterung der Kommunikation unter Wissenschaftlern.<sup>29</sup> Durch das Festsetzen einer charakteristischen Zahl für jeden Begriff versprach sich Leibniz Theorien operativ (messbar) machen zu können und Argumentationen in Zukunft so exakt wie arithmetische Beweise führen zu können.<sup>30</sup> Um 1680 drückte er die Hoffnung aus, wissenschaftliche oder philosophische Dispute in Zukunft rechnerisch lösen zu können:

*»Es wird dann beim Auftreten von Streitfragen für zwei Philosophen nicht mehr Aufwand an wissenschaftlichem Gespräch erforderlich sein als für zwei Rechnerfachleute. Es wird genügen, Schreibzeug zur Hand zu nehmen, sich vor das Rechengesetz zu setzen und zueinander (wenn es gefällt, in freundschaftlichem Ton) zu sagen: Lasst uns rechnen!«<sup>31</sup>*

In der *Characteristica Universalis* sollten sich nach Leibniz Philosophie und Mathematik zu einer formalisierten Wissenschaftssprache verbinden. Diese bereits im 17. Jahrhundert geäußerte Idee war radikal und revolutionär. Das hohe Ideal einer gänzlich formalisierten Sprache wurde von zahlreichen Gelehrten ausführlich untersucht. Eine Realisierbarkeit als Ganzes gilt mittlerweile als ausgeschlossen. Die Grenzen wurden bereits unter anderem durch Gödels *Unvollständigkeitssatz*<sup>32</sup> genau bestimmt. Dieser beweist, dass es selbst in hinreichend starken formalen Systemen, wie beispielsweise der Arithmetik, Aussagen gibt, die weder bewiesen, noch widerlegt werden können.

Immerhin konnte das Leibniz'sche Programm in der modernen Logik in begrenztem Umfang realisiert werden. In der Aussagenlogik werden zusammengesetzte Aussagen in einfachere Teilaussagen zerlegt. In Erweiterung dazu wird mit Hilfe der Prädikatenlogik die innere Struktur von Aussagen untersucht, die nicht mehr weiter zerlegt werden können. Die Prädikatenlogik wurde unter anderen von Gottlob Frege entwickelt und im Jahre 1879 in dessen *Begriffsschrift*<sup>33</sup> veröffentlicht. Dabei handelt es sich um ein logisches System, das es erlaubt, einen weiten Bereich wissenschaftlicher Argumente zu formalisieren und auf ihre Gültigkeit zu prüfen.<sup>34</sup> Der Begriff des Prädikats hat in der Logik eine etwas andere Bedeu-

<sup>27</sup> Primzahlen haben die besondere Eigenschaft nur durch 1 und sich selbst ganzzahlig teilbar zu sein.

<sup>28</sup> Um das Wort aussprechen zu können, sollten die Ziffern 1 bis 9 durch die lateinischen Konsonanten b, c, d, f, g, h, l, m, n ersetzt werden. Die Zahlenreihen (1, 10, 100, 1000, 10000) sollten durch die Vokale a, e, i, o, u bezeichnet werden. Würde ein Begriff mit der Zahl 81374 bezeichnet, so hätte er beispielsweise die Lautfolge „mubodilefa“ = 8(m)X10000(u) + 1(b)X1000(o) + 3(d)X100(i) + 7(l)X10(e) + 4(f)X1(a).

<sup>29</sup> Vgl. Blanke 1996, 31.

<sup>30</sup> Leibniz 1996, 21.

<sup>31</sup> Leibniz 1680, zit. n. Bauer 1992, 294.

<sup>32</sup> Vgl. Gödel 1931.

<sup>33</sup> Frege 1879.

<sup>34</sup> Vgl. Barwise/Etchemendy 2005, 2f.

tung als in der linguistischen Grammatik. Als Prädikat wird in der Logik eine Wortfolge mit Leerstellen bezeichnet, die sich in eine wahre oder in eine falsche Aussage verwandelt, sobald in jede Leerstelle ein Eigenname<sup>35</sup> eingesetzt wird. Die Wortfolge „... ist ein Mensch“ stellt ein Prädikat in diesem Sinne dar. Durch das Einsetzen von „Sokrates“ in die Leerstelle entsteht der (wahre) Aussagesatz „Sokrates ist ein Mensch“. Setzt man in die Leerstelle eines Prädikats statt eines Eigennamens eine Variable ein, entsteht eine sogenannte *Satzfunktion*.

$$F(x) = \text{„}x \text{ ist ein Mensch“}$$

$F(x)$  ist beispielsweise eine Funktion, die nur für die Eigennamen derjenigen Individuen, die Menschen sind, den Wahrheitswert *wahr* ausgibt und für alle anderen den Wahrheitswert *falsch*. Die Anzahl der Leerstellen eines Prädikats bezeichnet man als dessen *Stelligkeit*. Durch mehrstellige Prädikate können auch Relationen ausgedrückt werden. Das zweistellige Prädikat „... war ein Schüler von ...“ wird zur Satzfunktion  $F(x,y) = \text{„}x \text{ war ein Schüler von } y\text{“}$ . Beim Einsetzen von Platon für  $X$  und Sokrates für  $Y$  gibt diese Funktion beispielsweise den Wahrheitswert *wahr* aus. In der Logik wird die natürliche Sprache durch eine künstliche formale Sprache mit streng definierten Schlussregeln ersetzt. Als Lehre des vernünftigen Schlussfolgerns ist die Logik nicht nur ein wichtiger Bestandteil der Philosophie, sondern auch für die Mathematik und die Theoretische Informatik von großer Bedeutung.

## Formalisierte Entwurfsaufgaben

Im Jahre 1987 entwickelten Jon Barwise und John Etchemendy das Computerprogramm TARSKI'S WORLD, um Studenten den Umgang mit den Gesetzen der Logik auf spielerischem Weg zu vermitteln. Die beiden Mathematiker und Philosophen hatten die Idee einer räumlichen Umsetzung der Prädikatenlogik (Abb 0.2).

TARSKI'S WORLD besteht aus zwei Komponenten. Den ersten zentralen Bestandteil bildet ein Schachbrett, welches eine dreidimensionale Repräsentation eines im Grunde nur zweidimensionalen Datenmodells (Matrix) darstellt. In dieser Modellwelt können von der Spielerin oder dem Spieler unterschiedlich große geometrische Körper, wie beispielsweise Würfel oder Pyramiden platziert werden, und zur eindeutigen Identifizierung mit einem Buchstabe als Namen versehen werden.

Die zweite Komponente des Computerprogramms stellt ein Skriptfenster dar. Dieses erlaubt die Eingabe logischer Sätze im Sinne der Prädikatenlogik. Zur Beschreibung bestimmter Zustände der in der Welt abgesetzten Körper steht eine Reihe von Satzfunktionen zur Verfügung. Der Ausdruck  $\text{Cube}(b)$  steht hier beispielsweise für die Satzfunktion „ $b$  ist ein Würfel“ und beschreibt somit eine Eigenschaft von Objekt  $b$ .  $\text{Adjoins}(d,f)$  steht für „ $d$  und  $f$  sind benachbart“ und drückt somit eine Beziehung zwischen zwei Körpern aus. Abbildung 0.3 zeigt eine vollständige Auflistung der von TARSKI'S WORLD unterstützten Prädikate. Die Satzfunktionen können rekursiv verschachtelt und durch logische Operatoren (Junktoren)

---

<sup>35</sup> Als Eigenname kann neben Personennamen auch die Bezeichnung irgendeines Objekts, einer Zahl oder eines Umstandes dienen.

verknüpft werden. Der Ablauf des Spiels funktioniert folgendermaßen: In TARSKI'S WORLD erhält die Spielerin oder der Spieler eine „Entwurfsaufgabe“ in Form eines Skripts. Folgendes Beispiel zeigt vier Prädikate, welche durch den logischen Operator UND ( $\wedge$ ) verknüpft wurden:

$$Tet(a) \wedge Cube(b) \wedge Rightof(b,a) \wedge Smaller(b,a)$$

Frei übersetzt bedeutet dies: „Baue eine Welt, bestehend aus einem Tetraeder und einem Würfel, der zum einen größer als der Tetraeder und rechts neben diesem angeordnet sein sollte.“ Die Aufgabe der Spielerin oder des Spielers besteht dann darin, aus den zur Verfügung stehenden geometrischen Figuren eine Welt aufzubauen, auf welche möglichst alle, in der Aufgabenstellungen enthaltenen Teilaussagen, zutreffen, das heißt in welcher die logisch verknüpfte Gesamtaussagen wahr wird. Nachdem die Spielerin oder der Spieler die benötigten Körper auf dem Spielbrett angeordnet hat, wird die maschinelle Bewertung des Entwurfes durch das Betätigen der Schaltfläche *Verify* eingeleitet. Ist eine Aussage wahr, wird der entsprechende Satz automatisch grün markiert. Falsche Aussagen werden rot gekennzeichnet. Für jede Aussage über die Welt kommen nur zwei Zustände in Frage: Wahr oder Falsch.<sup>36</sup>

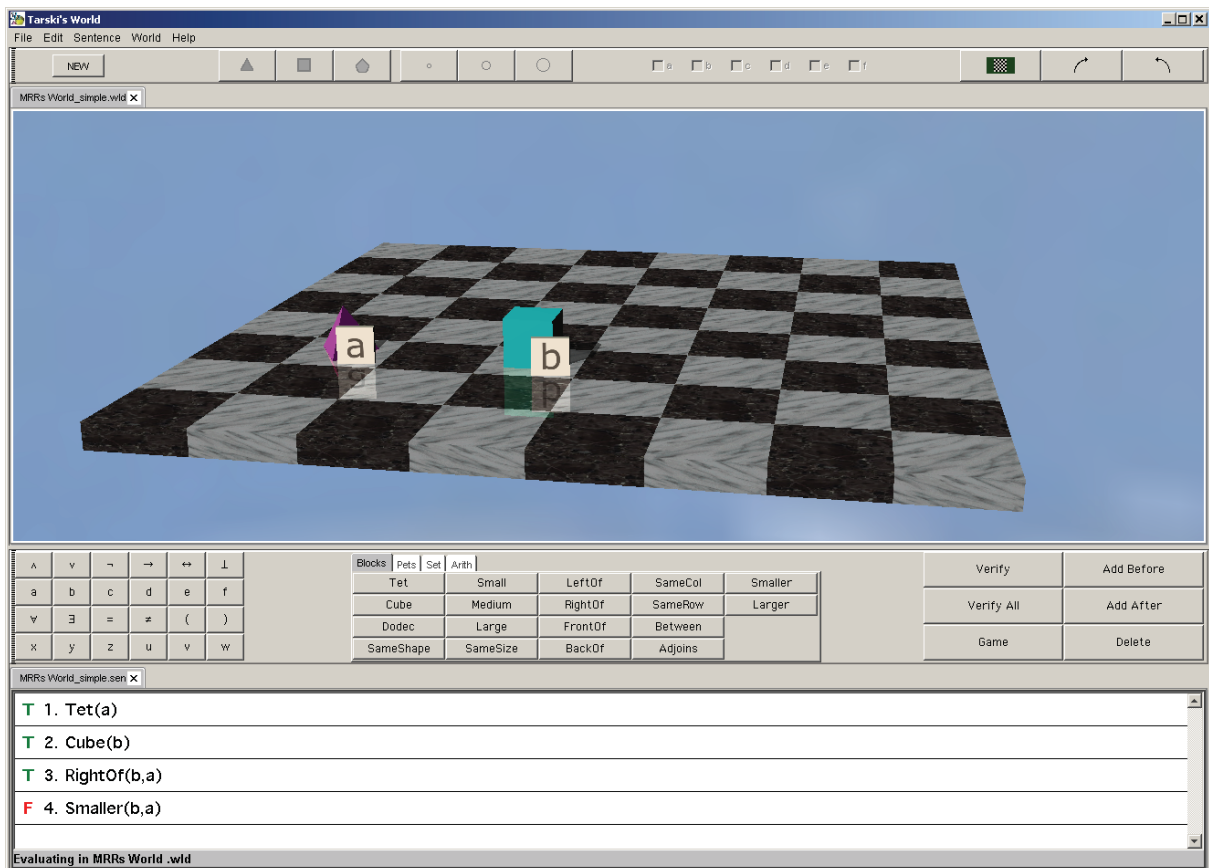


Abb 0.2 TARSKI'S WORLD 1987: Screenshot

<sup>36</sup> Vgl. Barker-Plummer u.a. 2008.

Tet(a)	a ist ein Tetraeder
Cube(a)	a ist ein Würfel
Dodec(a)	a ist ein Dodekaeder
Small(a)	a ist klein
Medium(a)	a ist mittelgross
Large(a)	a ist gross
SameSize(a,b)	a ist genauso groß wie b
SameShape(a,b)	a hat dieselbe Form wie b
Larger(a,b)	a ist grösser als b
Smaller(a,b)	a ist kleiner als b
SameCol(a,b)	a ist in derselben Spalte wie b
SameRow(a,b)	a ist in derselben Zeile wie b
Adjoins(a,b)	a und b befinden sich auf (nicht diagonal) benachbarten Quadraten
LeftOf(a,b)	a befindet sich dem linken Rand des Feldes näher als b
RightOf(a,b)	a befindet sich dem rechten Rand des Feldes näher als b
FrontOf(a,b)	a befindet sich der Vorderkante des Feldes näher als b
BackOf(a,b)	a befindet sich der Hinterkante des Feldes näher als b
Between(a,b,c)	a, b und c befinden sich in derselben Zeile, Spalte oder Diagonale, und a liegt zwischen b und c

Abb 0.3 TARSKI'S WORLD: Verfügbares Vokabular an Satzfunktionen

Das Besondere an den in TARSKI'S WORLD zur Verfügung gestellten Satzfunktionen besteht darin, dass diese tatsächlich als Computerroutinen implementiert wurden. Das heißt, der Wahrheitswert jeder einzelnen Aussage wird mit Hilfe von Algorithmen, welche die Datenstruktur analysieren, automatisch ermittelt. Jedes Prädikat verfügt dazu über eine Programmfunktion (Methode), der die benötigten Daten (A, B) in Form von *Argumenten* übergeben werden. Als *Resultat* wird ein Rückgabewert, im konkreten Fall eine boolesche Variable berechnet. In Abhängigkeit der Syntax der eingesetzten Programmiersprache könnte der Quellcode für eine solche Funktion beispielsweise folgendermaßen aussehen:

```
function Cube(A: object) : boolean;
begin
  if A.Objecttype=Cube then Result := true else Result:= false;
end;

function LeftOf (A,B: object) : boolean;
begin
  if A.XPos < B.XPos then Result := true else Result:= false;
end;
```

Abb 0.4 Delphi-Code, Messfunktionen der Prädikate *Cube* und *LeftOf*

Bei diesem Code-Beispiel wird davon ausgegangen, dass die Informationen über die geometrische Form und Position der Datenobjekte A und B am Spielfeld in den properties Objecttype beziehungsweise XPos und YPos gespeichert sind.

## Maschinelle Erkenntnis

Wie die meisten Spiele stellt auch TARSKI'S WORLD<sup>37</sup> ein ideales, hermetisch abgeschlossenes System dar. Das Spiel besteht aus einer Modellwelt und verfügt über eine formale Sprache mit eindeutiger Interpretation innerhalb dieser Welt. Dies bedeutet, dass für jedes Objekt definitiv feststeht, ob es die gewünschten Eigenschaften besitzt oder nicht, beziehungsweise ob es in der gewünschten räumlichen Beziehung zu anderen Objekten steht oder nicht. Das Programm ist in der Lage zu überprüfen, ob zum Beispiel das Objekt mit dem Namen a tatsächlich ein Würfel ist, oder ob Objekt b links oder rechts von Objekt a liegt. Vergleichbar mit einem Debugger<sup>38</sup> gibt das Programm Auskunft über die Wahrheit der Aussagen über die Modellwelt. In TARSKI'S WORLD werden Erkenntnisse, wie es im Grunde schon die Leibniz'sche Vision vorsah, ausschließlich durch Algorithmen gewonnen. Jedes Prädikat hat eine festgelegte Interpretation, die zu der entsprechenden, natürlichsprachigen Aussage der zugehörigen Satzfunktion passt. Das Urteil ob ein räumliches Arrangement den, in der formalen Sprache der Prädikatenlogik geäußerten Wünschen entspricht, wird in TARSKI'S WORLD ausschließlich nach den Gesetzen der Logik gefällt. Das bedeutet nicht, dass es in diesem Spiel für jede Entwurfsaufgabe nur eine oder „die“ richtige Lösung gibt. Neben dem in Abbildung 0.2 gezeigte Entwurf existiert eine große Anzahl anderer räumlicher Konstellationen, die sämtliche Prädikate erfüllen. Beispielsweise könnte der Würfel a auf eine beliebige Position rechts neben dem Tetraeder verschoben werden, ohne dass sich an dem Ergebnis etwas ändern würde.

Unabhängig von der angewandten Entwurfsmethode würde es Vorteile mit sich bringen, auch bei der Beurteilung von Architektur etwas mehr Kalkül anzuwenden. Mit dem Ausdruck Kalkül ist hier ein System aus Regeln gemeint, mit denen sich aus gegebenen Aussagen weitere Aussagen ableiten lassen. Dabei könnten ähnliche Prinzipien wie in TARSKI'S WORLD angewendet werden. Der amerikanische Architekt William J. Mitchell formulierte schon 1990 in seinem Buch *The Logic of Architecture* die Idee einer *kritischen Sprache* für die Beschreibung von Architektur in der Syntax der Prädikatenlogik.<sup>39</sup> Mitchell wies darauf hin, dass es möglich sei, Raumprogramme als Sammlungen von Gebäudemerkmalen aufzufassen, die in der Sprache der Prädikatenlogik verfasst sind. Als Beispiel führt er folgenden Teil eines solchen Programmes an, in dem die Anforderungen an eine zu planende Küche beschrieben werden:

»Larger (kitchen, 80) and Smaller (kitchen, 120)  
 Warmer (kitchen, 65) and Colder (kitchen, 75)  
 Ventilated (kitchen)  
 Daylight (kitchen)  
 Adjacent (kitchen, dining)«<sup>40</sup>

<sup>37</sup> Barwise/Etchemendy 2005, 21f.

<sup>38</sup> Als Debugger bezeichnet man ein Programm, das „Fehler [Anm. bugs] in der Programmierung sucht und ausschaltet“. Vgl. Duden online: Debugger, 22.10.2014.

<sup>39</sup> Vgl. Mitchell 1996, 199f.

<sup>40</sup> Mitchell 1996, 199.

Jedes der Prädikate beschreibt eine bestimmte Anforderung an den Funktionsbereich *Küche*. Die Aufgabe der Designerin oder des Designers liegt nun darin, möglichst alle dieser Zustände durch eine geeignete räumliche Anordnung zu erfüllen. Die Beurteilung einer Entwurfslösung besteht in der Folge darin, die Wahrheit der Prädikate innerhalb eines konkreten Gebäudeentwurfs festzustellen.

Bei der Übertragung des in TARSKI'S WORLD angewandten maschinellen Erkenntnisprinzips auf eine automatisierte Beurteilung von Entwurfslösungen, müssen die besonderen Eigenschaften, welche viele architektonische Entwurfslösungen auszeichnen, „einkalkuliert“ werden. Die erste Eigenheit besteht in dem Realitätsbezug architektonischer Modellwelten. Ein zweites Charakteristikum liegt in der Vagheit der auf diese Welt angewandten Sprache und in der damit verbundenen Unschärfe.

## Realitätsbezug

Die Objekte in Tarski's Modellwelt stehen in keiner Beziehung zur realen Welt. Die räumlichen Arrangements, die hier entwickelt werden, haben keinerlei Bedeutung für die Wirklichkeit. Das ist auch gar nicht notwendig, da das Computerprogramm ausschließlich dafür entwickelt wurde, Studierenden den Umgang mit dem System der Prädikatenlogik auf plastische Weise beizubringen. Die Semantik der Modelle, welche zur architektonischer Formfindung herangezogen werden, stellt sich dagegen anwendungsbedingt wesentlich komplexer dar: Architektonisches Entwerfen zielt in der Regel darauf ab, die reale Welt (baulich) zu verändern. Die räumlichen Gebilde, welche während des Entwurfsprozesses entwickelt werden, stehen für mögliche zukünftige Zustände der realen Welt.<sup>41</sup> Um die Prinzipien von TARSKI'S WORLD auf die Beurteilung von Gebäudeentwürfen zu übertragen, wird es notwendig sein, das für die Wirklichkeit bedeutungslose Spielbrett durch eine Modellwelt zu ersetzen, welche die besondere Eigenschaft der *Isomorphie zur Realität* aufweist (siehe Fußnote 23). Nur unter dieser Voraussetzung sind in der Folge nach der Analyse des Modells Rückschlüsse auf die Wirklichkeit möglich.

Auch architektonische Entwürfe werden heute schon größtenteils in dreidimensionalen digitalen Modellwelten entwickelt. Den momentanen Höhepunkt dieser Entwicklung stellt das *Building Information Modeling* (BIM) dar, welches im Wesentlichen auf das *Building Description System* (BDS) von Charles „Chuck“ Eastman aus dem Jahre 1974 zurückgeht.<sup>42</sup> Die Philosophie von BIM besteht darin, sämtliche Modellinformationen über ein Gebäude digital zu erfassen und in einem zentralen Datenmodell zu speichern. Bei der Modellwelt von BIM handelt es sich um den Versuch einer informationstechnischen 1:1 Abbildung von geplanter oder bereits gebauter Realität. Die Eigenschaften der Objekte in der Modellwelt stehen für Eigenschaften der entsprechenden Objekte in der realen Welt. Genauso stehen Beziehungen zwischen Objekten in der Modellwelt für Beziehungen zwischen realen Objekten.

Eastman sah, neben der Effizienzsteigerung bei der Erstellung von Konstruktionsplänen, einen der Hauptvorteile zentraler Gebäudedatenmodelle in der Möglichkeit der maschinellen Evaluierung. Ein frühes Beispiel der Umsetzung dieser Idee stellt der BUILDING DESIGN ADVISOR (BDA) dar. Diese am *Lawrence Berkeley National Lab* in der Mitte der

---

<sup>41</sup> Vgl. Mitchell 1996, 63f.

<sup>42</sup> Vgl. Eastman 1974, Eastman 1975.



1990er Jahre entwickelte Software benutzt ein spezielles objektorientiertes, assoziatives Gebäudemodell für die Simulation physikalischer Phänomene. Das System liefert Informationen über das zukünftige Verhalten eines Gebäudes und zeigt an, wie sich Veränderungen der Orientierung, der Geometrie oder der eingesetzten Materialien auf die Qualität einer Entwurfslösung auswirken. Das Programm bietet sogar einen einfachen Assistenten zur Optimierung eines Entwurfs. Die in BDA durchführbaren Analysen beschränken sich auf den thematisch eingegrenzten Bereich der Energieeffizienz und Belichtung von Gebäuden.<sup>43</sup>

Building Information Models sind durch ihre Isomorphie zur Realität sehr gut dafür geeignet, zu überprüfen, ob ein Gebäudeentwurf die im Raumprogramm beschriebenen funktionalen Anforderungen erfüllt. Trotz großer technologischer Fortschritte im IT-Bereich und der raschen Verbreitung des BIM Ansatzes, wurde die seit den 1970er Jahren prophezeiten Möglichkeit der maschinellen Evaluierung von Gebäuden bis heute nur wenig genützt. Aufgrund des ganzheitlichen Ansatzes sind BIM Systeme durch die zunehmenden technischen Anforderungen vor allem in der Phase der Ausführungsplanung zu sehr komplexen Datenstrukturen herangewachsen. Ein Großteil der Elemente und angebotenen Funktionen ist dabei für die frühe Entwurfsphase nicht von Bedeutung oder sogar eher kontraproduktiv. Die Denkmuster beim Entwerfen unterscheiden sich von denen des technischen Konstruierens. Für Gebäudeentwürfe in einer frühen Entwurfsphase, etwa im Detailgrad eines Architekturwettbewerbs - Maßstab 1:200 oder kleiner - wäre ein einfacheres Gebäudedatenmodell, dessen Entitäten sich semantisch mehr am entwerferischen Denken beziehungsweise an der Art, wie die funktionalen Anforderungen an Architektur in Raumprogrammen beschrieben werden, besser geeignet.<sup>44</sup>

## Vagheit

Der deutsche Designtheoretiker Horst W. J. Rittel bezeichnet architektonische Entwurfsaufgaben in seinem Aufsatz *Dilemmas in a General Theory of Planning* als *Wicked Problems*, also „boshafte Probleme“.<sup>45</sup> Nach Rittel gehört es unter anderem zum Wesen dieser Klasse von Problemen, nicht exakt definiert zu sein. Die Problemstellungen, mit denen Architektinnen und Architekten beim Entwerfen von Gebäuden konfrontiert werden, sind nicht nur wesentlich komplexer, sondern in der Regel auch viel unschärfer definiert als die abstrakten Designaufgaben in TARSKI'S WORLD. Für viele sprachlich formulierte Eigenschaften wie beispielsweise *gute Orientierung*, oder *ausreichende Belichtung*, die in Raumprogrammen von Räumen gefordert werden, existiert keine einheitlich, intersubjektiv festgesetzte Auslegung. Deshalb ist bei der Beurteilung eines vorliegenden Gebäudeentwurfs auch nicht immer klar, ob die Räume die gewünschten Eigenschaften erfüllen oder nicht. Genauso ist wenig sichergestellt, dass jeder dieselbe Vorstellung mit den benutzten Bezeichnungen verbindet.

Um unscharf definierte Merkmale in maschinenlesbare Prädikate umzuwandeln, wird es notwendig sein, die Bedeutungen der in heutigen Raumprogrammen benutzten alltags-

---

<sup>43</sup> Vgl. Building Design Advisor V3.0 - User's Guide; <http://gaia.lbl.gov/BDA/>

<sup>44</sup> Entgegen dieser Einschätzung werben die Anbieter führender BIM Plattformen gerade damit, dass ihre Programme der Denkweise von Architekten entsprechen.

<sup>45</sup> Vgl. Rittel/Webber 1973, 4.

sprachlichen Merkmalsbezeichnungen zu präzisieren, und die damit verbundenen Vorstellungen explizit festzulegen und zu vereinbaren.

Ähnlich verhält es sich mit der *Richtigkeit* einer architektonischen Entwurfslösung. Nach Rittel gibt es für Planungsaufgaben, wie man sie in der Architektur vorfindet, keine richtigen oder falschen Lösungen, sondern nur bessere und schlechtere. Ein Bewertungssystem, das nur die beiden Zustände *wahr* oder *falsch* kennt, kann dieser Gattung von Problemen nicht gerecht werden. Bei der Beurteilung von Gebäudeentwürfen genügt es nicht nur festzustellen, ob ein Entwurf das ihm zugrundeliegende Raumprogramm erfüllt oder nicht. Um beispielsweise mehrere Entwurfsvarianten miteinander vergleichen zu können, muss auch festgestellt werden, zu welchem Grad die definierten Ziele jeweils erfüllt werden. Nach dem *aristotelischen Zweiwertigkeitsprinzip*<sup>46</sup> kann eine Aussage nur wahr oder falsch sein. Anstatt der zweiwertigen Prädikatenlogik scheint hier die Anwendung eines Systems zielführend, welches auch in der Lage ist, mit Nuancen zwischen wahr und falsch umzugehen.

Abseits der klassischen zweiwertigen Logik existiert noch ein anderer Ansatz, der unter dem Namen Fuzzylogik bekannt wurde. Mit der Begründung der Fuzzylogik nahm der Informatiker Lotfi Zadeh<sup>47</sup> in der zweiten Hälfte des 20. Jahrhunderts intensiv die Beschäftigung mit dem Bereich zwischen wahr und falsch auf, dessen Existenz im Grunde schon Platon in der griechischen Antike postuliert haben soll.<sup>48</sup> Die Fuzzylogik wurde speziell für die Modellierung von Unsicherheit und Vagheit von umgangssprachlichen Beschreibungen entwickelt. Mit dieser Methode kann die sogenannte *Fuzziness* von graduellen Angaben wie *ein bisschen*, *ziemlich*, *stark* oder *sehr* in mathematischen Modellen erfasst werden. Diese *Logik unscharfer Begriffe* ist in vielen Anwendungen nützlich, insbesondere wenn von einem Sachverhalt oder einem Problem keine mathematische Beschreibung vorliegt, sondern nur eine verbale Schilderung. Mittels Fuzzylogik kann aus natürlichsprachigen Sätzen und sprachlich formulierten Regeln eine mathematische Beschreibung gewonnen und in Rechnersystemen sinnvoll genutzt werden.<sup>49</sup> In der Entscheidungstheorie werden *Fuzzy-Sets*<sup>50</sup> dafür eingesetzt, die Präferenzen eines *Entscheiders* graduell in Form von sogenannten Wertfunktionen abzubilden. Diese Technik kann auch für die Spezifikation der Zielvorstellungen an ein zukünftiges Gebäude herangezogen werden.

## Abgrenzung & Forschungsfragen

In der vorliegenden Arbeit wird der Frage nachgegangen, wie die in Raumprogrammen beschriebenen Gebäudemerkmale unter Einsatz geeigneter Computersoftware in einem Entwurf durch „Entdeckungsalgorithmen“, den im Zentrum der nachfolgenden Untersuchungen stehenden ARCHITEKTUR ROUTINE(n), erfasst werden können. In weiterer Folge wird erörtert, wie empirisch festgestellt werden kann, ob und zu welchem Grad ein Gebäudeentwurf eine Entwurfsaufgabe löst; wie weit ein Entwurf sozusagen „das Wort hält“.

---

<sup>46</sup> Durch den Satz vom ausgeschlossenen Dritten (*tertium non datur*) wurde von Aristoteles eine dritte Möglichkeit zwischen den kontradiktorischen Gegensätzen wahr und falsch kategorisch ausgeschlossen.

<sup>47</sup> Zadeh 1965.

<sup>48</sup> Vgl. Dilthey 1990, 196.

<sup>49</sup> Vgl. Kruse u.a. 1993, 1f.

<sup>50</sup> Als Fuzzy-Sets werden unscharfe Mengen bezeichnet, welche nicht durch die Objekte definiert werden, die Elemente dieser Menge sind (oder nicht sind), sondern über den Grad ihrer Zugehörigkeit zu dieser Menge. Vgl. Kruse u.a. 1993, 9f.

Eng damit verbunden ist die Frage, auf welche Art dazu die Zielvorstellungen an ein zukünftiges Gebäude formuliert beziehungsweise in welcher Sprache solche Spezifikationen verfasst werden können. Es wird der Versuch unternommen, eine neue Methode *sprachbasierten Data-Minings* für die Evaluierung von Gebäudeentwürfen in einem *Building Information Model* anzuwenden. Dabei steht nicht das logische System der Prädikatenlogik im Zentrum der Betrachtungen, sondern die Weiterentwicklung der Idee einer Implementierung „aktiver Satzfunktionen“ in Form von Computerroutinen für die Bewertung digital modellierter Gebäudeentwürfe. Die funktionale Auswertung informierter, in sich abgeschlossener Gebäudedatenmodelle stellt sich, trotz deren Komplexität, verglichen mit der Suche nach Informationsmustern im World Wide Web, als relativ überschaubar dar. Dennoch ist die Analogie zu dem Begriff des Data-Minings gut dafür geeignet, die in der vorliegenden Arbeit angewandte, ausschließlich auf Algorithmen basierende Operationalisierungs-Strategie von klassischeren Begriffen wie „Messung“ oder „Skalierung“ abzugrenzen. Abgesehen davon wird sich noch zeigen, welche Potentiale die in der Folge vorgenommenen (Binnen)Untersuchungen auch für eine Anwendung auf das *Big Game*<sup>51</sup> im *grenzenlosen Datenmeer* mit sich bringen.

Es soll ausdrücklich festgehalten werden, dass hierbei nicht die Intention besteht, allgemeine Maßstäbe oder Werte für die Beurteilung von Gebäudeentwürfen zu setzen. Offenheit und Adaptierbarkeit für individuelle Bedürfnisse stellen, trotz der notwendigen methodischen Strenge wichtige Grundprämissen der vorliegenden Arbeit dar. Auch soll nicht der Frage nachgegangen werden, ob oder wie Architektur von Computern erzeugt werden kann. Für die Untersuchungen ist es nicht relevant, wie ein Gebäudeentwurf entstand. Aus dem Themenkomplex der maschinellen Evaluierung von Entwurfslösungen resultieren Fragestellungen, welche sich in drei Hauptthemen einteilen lassen. Diese werden in der vorliegenden Arbeit jeweils im Rahmen eines Kapitels beantwortet:

- \* Wie können die bislang in natürlicher Sprache verfassten Raumprogramme mit zeitgemäßen Mitteln formalisiert werden, damit die darin enthaltenen Vor-Schriften nicht nur von Menschen, sondern auch von Maschinen gelesen und verarbeitet werden können?
- \* Wie sieht eine geeignete Modellwelt – Gebäudedatenmodell – aus, in der die in einer formalen Sprache beschriebenen Gebäudeeigenschaften maschinell evaluiert werden können?
- \* Wie kann eine vage Vorstellung (z.B. des Begriffs von „gut orientierten“ Wohnräumen) in eine mess- und berechenbare Angabe umgewandelt werden und in einen Algorithmus zur Erfassung (z.B. des Merkmals *Orientierung*) in einem geeigneten digitalen Gebäudemodell implementiert werden?

---

<sup>51</sup> Synonym für Hochseefischerei.

## Kapitelübersicht

### 1

Nachdem in der Einleitung Beweggründe und Motivationen für eine maschinelle Evaluierung architektonischer Entwurfslösungen dargelegt werden und die methodische Herangehensweise skizziert wird, beschäftigt sich das erste Kapitel mit den typischen Eigenheiten von Raumprogrammen, mit denen Architekten heute in der Praxis des Entwerfens konfrontiert sind. Hier soll das komplexe und vielschichtige Wesen architektonischer Aufgabenstellungen allgemein und grundsätzlich durchleuchtet werden. Nach der Identifizierung wesentlicher Komponenten von Raumprogrammen, widmet sich das Ende des Kapitels der Rolle von Organisationsdiagrammen an der Schaltstelle zwischen Text und Raum.

### 2

Im zweiten Kapitel werden einige, zum Verständnis dieser Arbeit notwendige, Grundprinzipien und Anwendungen der Topologie und Graphentheorie, in der Architektur und verschiedenen anderen Wissenschaftsdisziplinen vermittelt. Auch wenn die Lösung des Königsberger Brückenproblems in der Literatur schon hinlänglich beschrieben wurde, enthält es immer noch Lehren allgemeiner Natur, die es im Kontext dieser Arbeit wert sind im Sinne eines „Lernen von Königsberg“ hervorgehoben zu werden. Diese betreffen das Thema der Modellbildung, der Programmierung und das Wesen maschineller (algorithmischer oder operationalisierter) „Erkenntnisprozesse“. Zudem wird erörtert, welcher Bedeutung in diesem Zusammenhang dem Begriff des *Transzendentalen* nach dem Philosophen Immanuel Kant zukommt.

### 3

Mit dem dritten Kapitel beginnt der Hauptteil der vorliegenden Arbeit. Hier wird das Konzept einer formalisierten Gebäudespezifikation vorgestellt. Als Grundlage wird dazu neben einer Charakterisierung qualitativer und quantitativer Merkmale, die aus der deskriptiven Statistik stammende Einteilung in Skalenniveaus beschrieben, sowie der in der Entscheidungstheorie gebräuchliche Einsatz von Präferenzfunktionen vorgestellt. Danach erfolgt die Darstellung des im Zuge der Forschungsarbeiten entwickelten Konzepts der EXTENSIBLE SPACE ALLOCATION LANGUAGE (XSAL), der Vorschlag eines Standards zur strukturierten, menschen- und maschinenlesbaren Beschreibung von Gebäudespezifikationen auf Basis der Auszeichnungssprache XML. Das Kapitel endet mit der Vorstellung des DESIGN GOAL MANAGERS, eine im Zuge der Forschungstätigkeiten entwickelten Applikation zur grafischen Erstellung formalisierter Gebäudespezifikationen.

### 4

Im vierten Kapitel wird das Gebäudedatenmodell vorgestellt, welches für die Untersuchungen der vorliegenden Arbeit herangezogen wird. Nach einer kurzen allgemeinen Darstellung der historischen Entwicklung und des Wesens des *Building Information Modelings* wird das Computerprogramm DRAFTPAD vorgestellt, welches vom Autor ursprünglich zur Unterstützung der eigenen praktischen Entwurfstätigkeit konzipiert wurde. Dieses Zei-

chenprogramm basiert auf einem simplen, semantisch speziell auf die frühe Entwurfsphase zugeschnittenen *assoziativen Gebäudemodell*, welches auch die Modellwelt der nachfolgenden Untersuchungen liefert.

## 5

Das fünfte Kapitel beschäftigt sich mit der Frage, wie die in im DESIGN GOAL MANAGER formulierten Gebäudemerkmale in einem Gebäudedatenmodell maschinell erfasst werden können beziehungsweise welche Voraussetzungen dafür geschaffen werden müssen. Dazu wird die Methode der ALGORITHMISCHEN EXPLIKATION vorgestellt, ein Verfahren zur Umwandlung von Gebäudemerkmalen in mess- und berechenbare Funktionen. Diese werden in der vorliegenden Arbeit als ARCHITEKTUR ROUTINE(n) bezeichnet. Die wiederholte Anwendung dieses Verfahrens an beispielhaften Gebäudemerkmalen mündet schließlich in einen erweiterbaren Katalog aus implementierten ARCHITEKTUR ROUTINE(n), welcher dieser Arbeit im Anhang beigefügt wurde.

## 6

Das sechste Kapitel beschäftigt sich mit Anwendungen der ARCHITEKTUR ROUTINE(n) in der Praxis des Entwerfens, sowie im Kontext generativer Entwurfsprozesse. Es wird dargestellt, wie Entwerferinnen und Entwerfer durch die maschinelle Evaluierung ihrer Entwürfe bei ihrem alltäglichen Handwerk unterstützt werden können.

Der erste Teil des Kapitels beschreibt, wie die maschinell festgestellten Merkmalsausprägungen mathematisch verarbeitet, und mit, aus der multivariaten Entscheidungstheorie stammenden Methoden ausgewertet werden können, um schlussendlich zu einem empirischen Gesamturteil über die Fitness eines Gebäudeentwurfs zu gelangen.

Am Beispiel der vom Autor entwickelten Software ARCHILL.ES, eine Verknüpfung der Funktionen von DRAFTPAD und des DESIGN GOAL MANAGERS, wird danach gezeigt, wie eine Entwurfsumgebung als Schnittstelle zwischen Mensch und Maschine durch den Einsatz der ARCHITEKTUR ROUTINE(n) konkret aussehen kann.

## Conclusion

Als Schluss der vorliegenden Arbeit wird ein kritischer Ausblick auf die zahlreichen Anwendungsgebiete der ARCHITEKTUR ROUTINE(n) geboten. Dabei sollen nicht nur die Potentiale der im Zuge der Forschungstätigkeiten gewonnenen Erkenntnisse, Entwicklungen oder Erfindungen für manuelle und generative Entwurfsmethoden durchleuchtet werden, sondern auch aufgezeigt werden, welche gänzlich neuen Möglichkeiten durch eine maschinelle Evaluierung von Gebäudeentwürfen in der Architekturlehre geschaffen werden und welche anderen Einsatzbereiche darüber hinaus entstehen können.

# 1 RAUMPROGRAMME – ARCHITEKTUR À LA CARTE?

Die Planung eines Gebäudes beginnt in den meisten Fällen nicht mit dem Entwerfen, sondern mit einer Diskussion über Wünsche und Zielvorstellungen. Die Ergebnisse dieses Prozesses werden heute in der Regel schriftlich in Form von Raumprogrammen festgehalten. Die ersten dokumentierten Ansätze solcher Programme finden sich schon in Vitruvs *Zehn Büchern über Architektur*<sup>1</sup> aus dem 1. Jahrhundert vor Christus und wurden in der Frührenaissance von dem Humanisten Alberti (1404 -1472) noch differenzierter beschrieben. Albertis Hauptwerk *de re aedificatoria*<sup>2</sup> richtete sich dabei weniger an Architekten als an gebildete Bauherren. Bis hin zur Moderne wurde Raumprogrammen ein stetig wachsender Stellenwert beigemessen. Mit dem im Jahre 1720 in Frankreich an der *Académie Royale d'Architecture* begründeten *Grand Prix de Rome*, einem jährlich ausgelobten Architekturwettbewerb, entwickelten sich Raumprogramme zu immer präziseren und strengeren Vorgaben. Durch das Programm des *Grand Prix de Rome*, welches von Jahr zu Jahr verschieden war, wurde jeweils ein Gebäudetyp mitsamt der darin aufzunehmenden Aktivitäten und räumlichen Anforderungen beschrieben. Dabei handelte es sich um ein förmliches Dokument, welches den Teilnehmern des Wettbewerbes in den frühen Jahren nur diktiert, und später auch in schriftlicher Form übergeben wurde. Wettbewerbsbeiträge, die den Vorgaben des Programms nicht Folge leisteten wurden aus dem Wettbewerbsverfahren ausgeschlossen.<sup>3</sup>

*»In conjunction with the great academic texts, such as those of Blondel, Durand, and later Guadet, the Grand Prix competition firmly established the idea of a formalized architectural program, against which designs could be checked for compliance.«<sup>4</sup>*

Die Programme des *Grand Prix de Rome* verfügten über einen strikten dreiteiligen Aufbau. Auf einen kurzen einleitenden Absatz folgte der Haupttext, in welchem die Anforderungen aller räumlichen Elemente eines zu entwerfenden Gebäudes detailliert beschrieben wurden. Einen wichtigen Teil bildete dabei eine hierarchische Aufstellung der für die zukünftige Nutzung des Gebäudes notwendigen Räume. Im letzten Abschnitt wurden schließlich allgemeine Vorgaben bezüglich der Größenordnung des Bauplatzes und des zukünftigen Gebäudes sowie über die einzureichende Anzahl von Plänen in vorgegebenen Maßstäben gemacht.<sup>5</sup>

---

<sup>1</sup> Vgl. Reber 2009, 279. (Buch 6 über Privatgebäude)

<sup>2</sup> Vgl. Theuer 1912.

<sup>3</sup> Vgl. Mitchell 1996 [erstveröffentlicht 1990] , 199.

<sup>4</sup> Mitchell 1996, 199.

<sup>5</sup> Vgl. Levine 1982.

Noch heute werden Architekturwettbewerbe im Grunde nach diesem Modell abgehandelt. Im Zuge der Projektentwicklung wird ein Programm formuliert, welches die Eigenschaften sowie die funktionalen Anforderungen an ein zu planendes Gebäude beschreibt und in weiterer Folge als Grundlage für die Erstellung eines räumlichen Gebäudeentwurfs dient. In den USA wird der meist partizipative Prozess der Erhebung solcher Nutzungsanforderungen an ein zukünftiges Gebäude als *programming* (brit. briefing) bezeichnet. „*If programming is problem seeking, then design is problem solving*“<sup>6</sup>, heißt es in William Peñas und Steven Parshalls im Jahre 1969 erstmals fibelartig erschienenen Buch *Problem Seeking – An Architectural Primer*, der ersten umfassenden Abhandlung zum Thema der nutzungsspezifischen Bedarfsplanung. Beim *Programming*<sup>7</sup> sollen die Eigenschaften und funktionalen Anforderungen eines zukünftigen Gebäudes als dessen Leistungen erfasst werden. Ein solches performatives Konzept schlägt auch die ISO 9699, *Performance standards in building – Checklist for briefing – Contents of brief for building design*, vor.<sup>8</sup> Diese Norm legt keine bestimmten Methoden oder Verfahren für die Bedarfsplanung fest. In schriftlicher Form sollen die Anforderungen an ein zu planendes Gebäude in einer vorgegebenen, gegebenenfalls zu modifizierenden Gliederung erfasst und durch Prüflisten auf ihre Vollständigkeit kontrolliert werden. Bei der Projektentwicklung sollen die qualitativen und quantitativen Ziele und Wünsche des Bauherrn im sogenannten *Nutzerbedarfsprogramm* festgehalten werden, welches in späteren Planungsphasen seine Fortschreibung im sogenannten *Raumbuch* findet. Mit diesem Planungswerkzeug kann eine Messlatte der Projektziele geschaffen werden, welche verbindliche Auskunft darüber geben soll, ob und inwieweit die Projektziele mit den Planungsergebnissen erfüllt wurden.<sup>9</sup>

Ein solches Programm beschreibt das Feld der Probleme a priori, ohne bereits formale Lösungsansätze zu liefern. Der Freiheitsgrad ist in dieser Phase eines Projekts noch nahezu uneingeschränkt. Als reines Gedankengebäude unterliegt ein Entwurfsprogramm noch keinen räumlichen, geometrischen oder physikalischen Zwängen. Das Raumprogramm ist eine Art ideales Repertoire, in dem alle Elemente eines zukünftigen Gebäudes noch scheinbar konfliktfrei nebeneinander existieren. Es beschreibt Besonderheiten, die einer Entwurfsaufgabe eigen sind, und gibt Auskunft über die Struktur, das heißt die Elemente und deren Beziehungen eines zukünftigen Gebäudes.

*»Die Struktur eines Bereichs freizulegen, heißt, eine ganze Virtualität der Koexistenz zu bestimmen, die vor den Wesen, den Gegenständen und den Werken dieses Gebietes existiert. [...] Was besteht in der Struktur nebeneinander? Alle Elemente, Verhältnisse und Verhältniswerte, alle Besonderheiten, die dem betrachteten Bereich eigen sind. Eine solche Koexistenz impliziert keinerlei Konfusion, keinerlei Unbestimmtheit: es sind differenzielle Verhältnisse und Elemente, die in einem vollkommen und vollständig bestimmten Ganzen nebeneinander bestehen.«<sup>10</sup>*

---

<sup>6</sup> Peña/Parshall 2001, 15.

<sup>7</sup> Der Begriff *Programming* wurde vom deutschen Architekten Gunter Henn in den 1980er Jahren in Europa eingeführt.

<sup>8</sup> Die ISO 9699 wurde im Jahre 1996 mit geringen Änderungen in die deutsche DIN 18205 übernommen.

<sup>9</sup> Vgl. AHO 2004.

<sup>10</sup> Deleuze [erstveröffentlicht 1969] 1992, 28.

Während des Entwurfsprozesses wird das virtuelle Begriffsgebäude in einen konkreten, räumlich ausgedehnten Gebäudeentwurf umgesetzt, welcher durch Pläne, analoge beziehungsweise digitale Modelle repräsentiert wird. Der Entwurfsprozess entspricht einem Aktualisierungsprozess, welcher laut Deleuze immer eine innere Zeitlichkeit impliziert.

Architekturwettbewerbe stellen ein wichtiges Anwendungsfeld für Gebäudespezifikationen dar. Den Kern vieler Wettbewerbsausschreibungen bildet eine tabellarische Zusammenstellung der funktionalen Anforderungen, welche die Fakten, der im Ausschreibungstext ausführlich (narrativ) beschriebenen architektonischen Aufgabenstellung, übersichtlich darlegt. Abbildung 1.1 zeigt beispielhaft das Raumprogramm eines Kindergartens. Jede Zeile der Tabelle beschreibt einen bestimmten Funktionsbereich. Die Zuordnung zu Obergruppen markiert thematische oder funktionale Einheiten. Zur eindeutigen Identifikation sind den Funktionsbereichen Raumnummern zugeordnet. In den einzelnen Spalten werden unterschiedliche Merkmale jedes Funktionsbereichs beschrieben.

Raumprogramme sind so unterschiedlich, wie die Problemstellungen die sie aufwerfen. Einige grundlegende Elemente finden sich jedoch in fast allen Programmen dieser Art. Die meisten Raumprogramme enthalten eine Auflistung der für ein zukünftiges Gebäude notwendigen Funktionsbereiche und charakterisieren diese durch spezifische Merkmale. Außerdem geben die meisten Raumprogramme Auskunft über notwendige Beziehungen dieser Bereiche untereinander. In den folgenden Unterkapiteln soll auf die grundlegenden Komponenten von Raumprogrammen näher eingegangen werden.

1 RAUMPROGRAMM				
Funktionseinheit	Anzahl	NF in m2		Bemerkungen
<b>1. Gruppeneinheiten</b>				
1.1. Gruppenraum	8	75	600	
1.2. Garderoben	8	42	336	Zugang zu Gruppenraum, Garderobe, Garten 1 Garderoberaum für 2 Gruppen möglich
Sanitärraum				Zugang zu Gruppenraum und Garderobe
Abstellraum				Zugang vom Gruppenraum
			<b>936</b>	
<b>2. Verwaltung</b>				
2.1. Kanzlei	1	21	21	in Eingangsnähe
2.2. Arzttraum/Therapieraum	1	21	21	Verbindungstüre zu Kanzlei
2.3. Besprechungsraum	1	21	21	
2.4. Sozialraum	1	21	21	Verbindungstüre zum Besprechungsraum
			<b>84</b>	
<b>3. Sonstiger Bereich</b>				
3.1. Küchenbereich	2	63	126	Küche, Speis, Bistrobereich
3.2. Waschküche	1	15	15	
3.3. Personalgarderobe	1	21	21	
3.4. Personalwaschraum	1	6	6	Zugang zum Personalraum
3.5. Personal-WC	4	3	12	2/Geschoss
3.6. Behinderten-WC	2	4	8	1/Geschoss
3.7. Putzmittelraum	2	6	12	1/Geschoss
3.8. allgemeiner Abstellraum	2	15	30	in Keller möglich
3.9. Kinderwagenabstellraum	1	30	30	in Eingangsnähe
			<b>260</b>	
<b>4. Kreativbereich</b>				
2.2. Snozelenraum	1	15	15	Entspannungsraum
2.3. Atelier (Wintergarten)	1	42	42	inkl. Materialraum
2.4. Konstruier- und Baubereich	2	15	30	1/Geschoss
			<b>87</b>	
<b>5. Spielhaus (Mehrzweckraum)</b>				
5.1. Gruppenraum, Garderobe, Sanitärraum	1	100	100	Zugang für externen Betrieb und interne Anbindung zum Kindergarten
			<b>100</b>	
<b>NUTZFLACHE- NF</b> gesamt				
Kindergarten			<b>1.467</b>	

Abb.1.1 Raumprogramm für den Entwurf eines Kindergartens



## 1.1 Funktionsbereiche

Eine grundlegende Eigenschaft jedes Gebäudes ist die Differenzierung von Raum. Beim Entwerfen eines Gebäudes wird Raum gegliedert. Architektur trennt Räume durch Elemente wie Wände oder Decken und ist in der Lage diese wiederum kontrolliert zu verbinden. Dabei werden Bereiche geschaffen, die verschiedene Aufgaben zu erfüllen haben. Entwurfsprogramme enthalten eine Auflistung solcher benötigter Funktionsbereiche. Je differenzierter die Gliederung der Flächenaufstellung ist, desto detaillierter können in der Folge auch die Zusammenhänge zwischen den einzelnen Bereichen dargestellt und beschrieben werden. Im Falle einer Bibliothek zum Beispiel lässt sich der *allgemeine Freihandbereich* in Unterzonen wie *Sachliteratur*, *Schöne Literatur* etc. weiter unterteilen. Ebenso kann ein Wohnraum in Bereiche zum Essen, Kochen, Sitzen etc. gegliedert werden. Funktionsbereiche können ihrer Nutzung entsprechend entweder separate Räume, oder Teilbereiche eines Raumkontinuums sein. Über das Raumprogramm kann beispielsweise definiert werden, ob eine Küche als eigener Raum oder als Teilbereich eines anderen Raums, wie beispielsweise eines Wohnzimmers, ausgebildet werden soll. Ebenso können Parkplätze und Fahrspuren als Bereiche einer Tiefgarage betrachtet werden, oder Blumenbeete als Bereiche des Gartens etc.. Befinden sich in einem Raumkontinuum mehrere Funktionsbereiche, kann es durchaus zu Nutzungs- beziehungsweise Flächenüberschneidungen kommen. Erst wenn ein Bereich begrifflich erfasst ist, wird es möglich, sich auf diesen sprachlich zu beziehen, dessen Merkmale zu beschreiben und mit anderen Bereichen in Beziehung zu setzen.

## 1.2 Merkmale

Die erste Auskunft über einen Funktionsbereich gibt dessen Bezeichnung. Die dafür gewählten Begriffe lassen in der Regel bereits auf die zukünftige Nutzung schließen. Raumbezeichnungen stellen ein komplexes Merkmal dar. Schon Aristoteles soll in seiner Schrift *De Anima* bemerkt haben, dass ein physisches Objekt nicht nur durch seine Form und Materialität, sondern genauso gut durch die Spezifikation seiner Funktion beschrieben werden kann.<sup>11</sup> Die Benennung der Funktion eines Raumes impliziert häufig eine Reihe unausgesprochener Eigenschaften. Wird ein Raum beispielsweise als „Wohnzimmer“ bezeichnet, so erhält dieser dadurch nicht nur einen Namen. Die Bezeichnung der Funktion eines Raums impliziert auch häufig eine ganze Reihe von assoziierten Eigenschaften. Beim Entwerfen dieses Raums, wird das Raumprogramm durch Erfahrungen oder angeeignetes Wissen ergänzt werden müssen. Dabei könnten unterschiedliche Schlüsse gezogen werden: Ein Wohnzimmer dient dem Wohnen, sollte also natürlich belichtet sein, vorzugsweise nach Süden oder Westen orientiert sein, zentral liegen, vielleicht 30m<sup>2</sup> groß sein, eventuell rechte Winkel aufweisen, um gut möblierbar zu sein etc.. Eine solche subjektive Sammlung von Merkmalen kann natürlich aufgrund geografischer, kultureller und vieler anderer Einflussfaktoren stark variieren und in der Folge zu unterschiedlichen Entwurfsergebnissen führen.

Andere Anforderungen an ein Gebäude, die häufig nicht explizit in Raumprogrammen festgehalten werden, entspringen Normen und Gesetzestexten. Die regional variierenden

---

<sup>11</sup> Vgl. Mitchell 1996, 183.

Bauordnungen enthalten Richtlinien und Gesetze, welche sich wesentlich auf den Entwurf auswirken können. Dies beginnt schon im städtebaulichen Maßstab mit Bebauungsdichteverordnungen oder Abstandsregeln und setzt sich im Inneren von Gebäuden in Form von Fluchtweglängen, Mindestbelichtungsflächen oder Barrierefreiheit etc. fort. Genauso stellen baukünstlerische Zielvorstellungen, wie zum Beispiel die Idee eines scheinbar schwebenden Baukörpers, eines gut proportionierten Gebäudes, oder aussergewöhnliche Ideen, wie die eines Gebäudes in Form einer Ente<sup>12</sup> in diesem Sinne Ergänzungen des Programms dar. Der Philosoph Nelson Goodman unterstreicht in seinem Aufsatz *How Buildings Mean* die Wichtigkeit der Symbolik in der Architektur mit folgenden Worten:

*»A building is a work of art only insofar as it signifies, means, refers, symbolizes in some way.«<sup>13</sup>*

Aus der Funktion jedes einzelnen Bereichs resultiert ein bestimmter Flächenbedarf. Ein zentraler Aspekt von Raumprogrammen besteht in der Ausweisung der für jeden Funktionsbereich benötigten Nutzfläche. Zur Nutzfläche zählen alle Flächen, welche dem Verwendungszweck eines Gebäudes unmittelbar dienen, wie beispielsweise Wohnen, Arbeiten, Bildung oder Freizeit. Ebenfalls zählen untergeordnete Flächen wie Sanitärräume, Müllräume oder Abstellräume zu dieser Flächenkategorie. Verkehrsflächen werden in Raumprogrammen selten definiert. Die zur Erschließung der Funktionsbereiche benötigten Flächen hängen in vielen Fällen stark von der Wahl der Typologie des späteren Gebäudeentwurfs ab und kann zum Zeitpunkt der Erstellung des Raumprogramms nur über Kennwerte, die aus ähnlichen Bauaufgaben gewonnen wurden, hypothetisch angesetzt werden.

Ein Funktionsbereich mit der Bezeichnung „Schlafzimmer“ und der gewünschten Fläche von zum Beispiel 15m<sup>2</sup> ist schon relativ präzise charakterisiert. Über die Angabe der Ausstattungsobjekte des Raumes können noch genauere Auskünfte über die notwendige Form eines Raumes beschrieben werden. Ein Schlafzimmer könnte mit einem Doppelbett oder nur mit einem Einzelbett möbliert sein, möglicherweise auch zusätzlich mit einer Schrankwand. Derartige Informationen wirken sich auf den späteren Gestaltungsspielraum beim Entwurf des Raumes aus. Bei den meisten Objekten ist der benötigte Raum wesentlich größer, als das Objekt selbst. Man denke hierbei beispielsweise an einen Billardtisch oder einen Klavierflügel. Je nach Funktion des Objektes besteht unterschiedlicher Bedarf an angrenzender Manipulations- oder Abstandsfläche. Es kann vorkommen, dass sich durch die Beschreibung der Ausstattung eine Flächenangabe sogar erübrigt. Manche Funktionsbereiche müssen als Ganzes eine bestimmte Größe und Form aufweisen. Dies ist zum Beispiel bei genormten Sportfeldern oder Parkplätzen der Fall. Genauso wie die Exponate eines Museums etwa die Höhe der Ausstellungsräume bestimmen können, resultiert die Geometrie

<sup>12</sup> *The Big Duck* ist ein Gebäude, welches um 1930 in Flanders, New York auf Long Island von dem Entenfarmer Martin Maurer errichtet wurde. Bei dem Gebäude handelt es sich um ein Geschäft zum Verkauf von Enten und Enteneiern. Das Bauwerk selbst hat seiner Funktion entsprechend die Form einer riesigen Ente. Das Gebäude war in den 60er und 70er Jahren des 20. Jahrhundert zahlreicher Kritik ausgesetzt, hatte aber auch architektonische Verteidiger: Robert Venturi beispielsweise empfand die Kombination von Form und Symbolik des Gebäudes als bemerkenswert. Von ihm wurde der Ausdruck „duck“ als allgemeine Bezeichnung für Gebäude geprägt, deren Architektur von symbolischen Aspekten dominiert wird. Vgl. [bigduck.org](http://bigduck.org).

<sup>13</sup> Goodman 1985, zit. n. Ernst u.a. 2009, 325.

eines Konzertsahls häufig aus akustischen oder visuellen Notwendigkeiten. Die Form eines Funktionsbereichs ist selten beliebig. Zu große, zu kleine oder ungünstig geformte Bereiche können sich negativ auf die Nutzung eines Gebäudes auswirken.

Nicht nur Ausstattungsobjekte, sondern auch Handlungsabläufe können Form und Lage der Funktionsbereiche eines zukünftigen Gebäudes indirekt vorgeben. Die in einem Gebäude stattfindenden Aktivitäten stellen die beteiligten Funktionsbereiche und Einrichtungsgegenstände in eine funktionale Beziehung. Abbildung 1.2 zeigt schematisch die Prozessketten zwischen einer Küche und einem Speiseraum. Im linken Diagramm führen schlecht angeordnete Küchengeräte und Funktionsbereiche zu konfusen sich selbst überschneidenden Handlungsabfolgen, sowohl beim Anrichten (A) als auch beim Abservieren (B) der Speisen. Rechts werden durch ein passendes Arrangement derselben Elemente wesentlich geradlinigere und einfachere Handlungsabläufe erzielt.

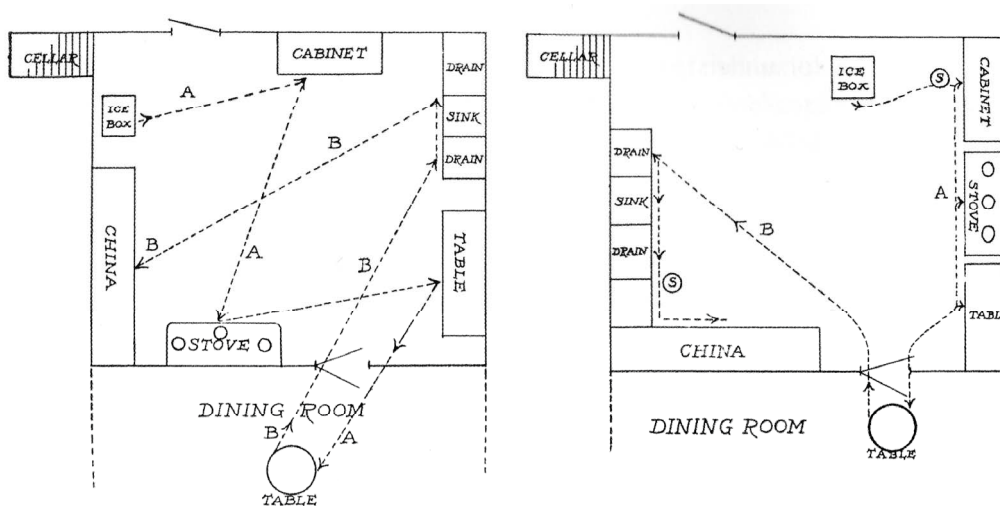


Abb 1.2 Christine Frederick 1914, Prozesskette zwischen Küche und Speiseraum.

Viele Merkmale von Räumen können ausschließlich im Zusammenspiel von Funktion, Ausstattung und den nutzungsbedingten, in einem bestimmten Funktionsbereich stattfindende Prozessen betrachtet werden. Merkmale wie beispielsweise die „Geräumigkeit“ eines bestimmten Raumes kann erst beurteilt werden, wenn auch genau geklärt ist, welche Tätigkeiten in dem betreffenden Raum stattfinden sollen und welche Objekte hierzu notwendig sind. Ein 300m<sup>2</sup> großes Foyer eines Museums könnte genauso als geräumig bezeichnet werden wie ein 5 m<sup>2</sup> großes Badezimmer, wenn auf die Badewanne verzichtet wird. Daraus ist ersichtlich, dass manche Merkmale relativer Natur sind. Geräumigkeit beispielsweise liegt im Bereich des Verhältnisses von „für die Funktion notwendiger Fläche“ zu „tatsächlich zur Verfügung stehender Fläche“. Ähnliches gilt auch für Merkmale wie die Belichtung. Ein „heller“ Erschließungsgang muss nicht gleichzeitig auch hell genug für eine Wohnnutzung sein. Neben den bereits erwähnten Eigenschaften können in Raumprogrammen für die einzelnen Funktionsbereiche eines zukünftigen Gebäudes beliebige Wunschmerkmale definiert wer-

den. Je mehr Merkmale aufgezählt werden, desto präziser werden die einzelnen Bereiche charakterisiert und unterscheidbar gemacht. In Summe dienen die Merkmale als Entwurfsparameter.

### 1.3 Beziehungen

Viele Funktionsabläufe innerhalb eines Gebäudes finden nicht in einem einzelnen Raum alleine statt. Aus diesem Grund können manche Anforderungen an ein Gebäude nicht durch die isolierte Spezifikation der Funktionsbereiche beschrieben werden. Als Vorgabe zur Anordnung der Räume ist hierfür eine Darstellung der wechselseitigen Beziehungen der einzelnen Bereiche notwendig. Dadurch wird die Stellung der Räume zueinander definiert. In dem Raumprogramm auf Abbildung 1.1 erfolgt dies in der Spalte *Bemerkungen* auf der rechten Seite der Tabelle. Hier wird beispielsweise ein Zugang vom Abstellraum zum Gruppenraum gefordert. Abbildung 1.3 zeigt eine Beziehungsmatrix aus Godfrey Thompsons Buch *Library Buildings*<sup>14</sup>, welche die Naheverhältnisse zwischen den einzelnen Räumen aufgrund der funktionalen Zusammenhänge einer mittelgroßen Bibliothek beschreibt. Die Bewertung der Beziehungen erfolgt in diesem Beispiel in fünf Abstufungen durch die Buchstabensymbole A bis E. Der Buchstabe A steht dabei etwa für „möglichst nahe“, E für „möglichst getrennt“. Die Buchstaben B bis D markieren Nuancen zwischen diesen beiden Extremen. Aus einer solchen Bewertung der Beziehungen resultiert eine sukzessive programmatische Vernetzung des Raumprogramms. Bei mehrgeschoßigen Gebäuden bekommt das Naheverhältnis von Bereichen eine zusätzliche dritte Dimension. Raumprogramme können beispielsweise den Hinweis enthalten, dass sich zwei bestimmte Funktionsbereiche auf demselben Höhenniveau befinden sollen. Auf diese Weise kann auch eine Definition der dreidimensionalen Konstellation der benannten Bereiche erfolgen.

Bei einer zunehmenden Anzahl von Funktionsbereichen oder anderer Komponenten einer Entwurfsaufgabe steigt die Anzahl der Beziehungen exponentiell und die Analyse kann schnell eine hohe Komplexität erlangen. In der *Site Selection Study of the Metropolitan Toronto Library Board*<sup>15</sup> aus dem Jahre 1971 wurde der Grad der Verflechtung sämtlicher Komponenten einer zukünftig zentralen Bibliothek in einer Beziehungsmatrix aufgezeichnet. Bei den 99 in dieser Studie identifizierten Grundkomponenten ergeben sich schon mehrere tausend Beziehungen.<sup>16</sup>

Viele architektonische Besonderheiten resultieren aus dem Zusammenspiel und der Wechselwirkung der einzelnen Gebäudeteile. Abgesehen von pragmatischen Faktoren wie kurze Weglängen in Krankenhäusern oder kreuzungsfreie Wege in Gastronomiebetrieben, können dabei genauso atmosphärische Aspekte im Vordergrund stehen. Der von Le Corbusier geprägte Begriff der *promenade architecturale* beispielsweise steht für die Folge der Bilder, welche sich einem Betrachter beim Durchschreiten eines Bauwerks ergibt. Durch die Verschränkungen von Innen- und Außenraum sowie das Ineinanderfließen der Räume durch den *Plan Libre* bildet Architektur den Raum von Bewegungsabläufen.<sup>17</sup>

<sup>14</sup> Vgl. Thompson 1989.

<sup>15</sup> Vgl. Moriyama 1971.

<sup>16</sup> Genaugenommen sind es ohne selbstreferenziellen Relationen  $(99 \times 99 - 99) / 2 = 4851$  Beziehungen.

<sup>17</sup> Vgl. Samuel 2010.



als Kreise eingezeichnet. Feine Linien markieren funktionale Zusammenhänge. Die einzelnen Kreise sind mit dem Namen des betreffenden Funktionsbereiches versehen und weisen unterschiedliche Farben auf. Manche der Kreise sind wiederum in zwei Hälften geteilt, was möglicherweise auf einen Zusammenschluss zweier Funktionen hindeutet. Der unterschiedliche Durchmesser der Kreise gibt Auskunft über den jeweiligen Flächenbedarf.

Mit Hilfe geeigneter Diagramme ist es möglich, eine Überleitung von sprachlich formulierten Raumprogrammen hin zu einem räumlich ausgedehnten Gebäudeentwurf einzuleiten. Das Diagramm ist in der Lage den Wechsel vom Text zum Raum einzuleiten und die topologische Struktur einer Entwurfsaufgabe zu enthüllen. Bei Organisationsdiagrammen dieser Art, handelt es sich, im Gegensatz zu anderen diagrammatischen Darstellungen, um visuelle Hilfsmittel in einer rein programmatischen Verwendung. Von Form oder Gestalt wird so weit wie möglich abstrahiert. Obwohl solche Strukturdiagramme bereits eine zweidimensionale Ausdehnung suggerieren, handelt es sich dabei im Gegensatz zu maßstäblichen Skizzen oder Plänen nicht um Abbilder einer räumlichen Vorstellung oder eines geometrischen Entwurfes, sondern um eine abstrakte Darstellung topologischer Zusammenhänge.

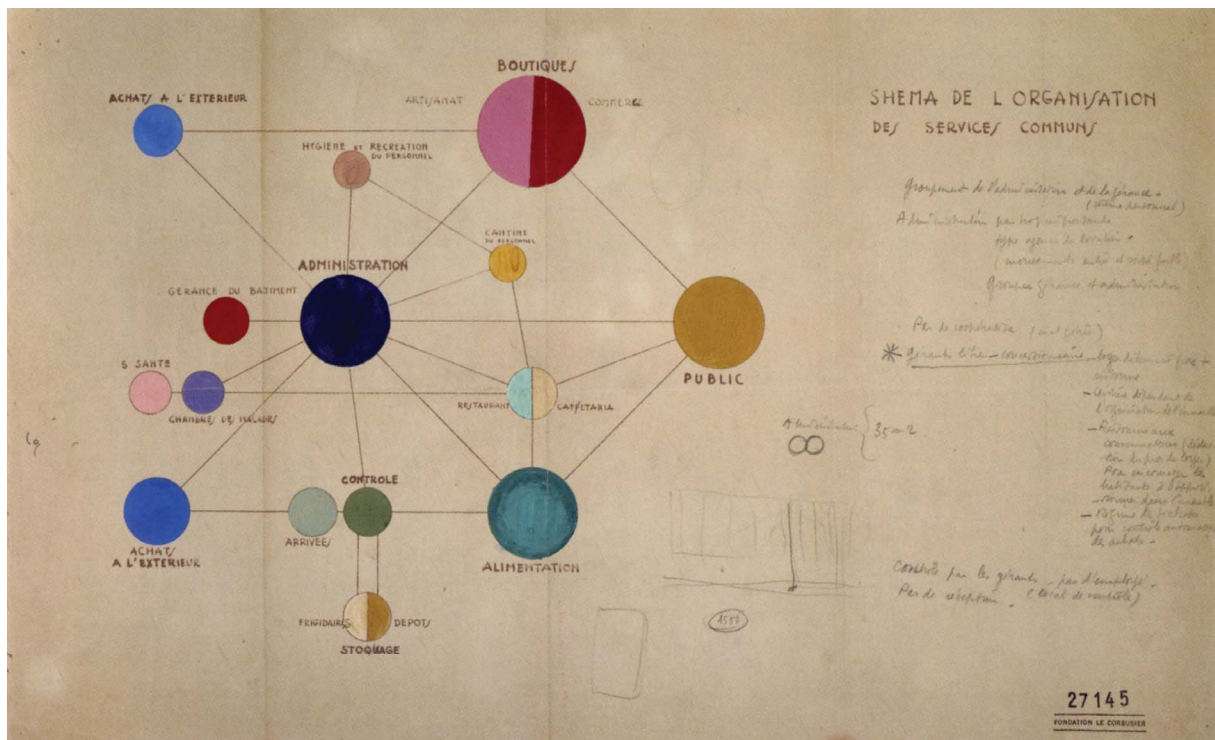


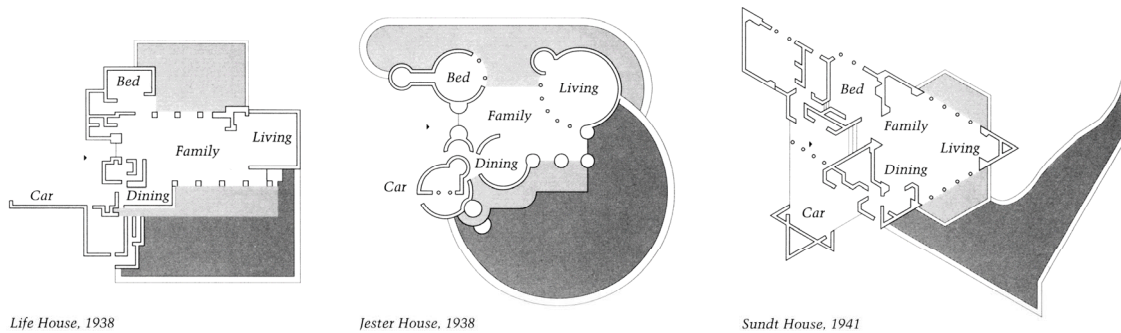
Abb 1.4 Le Corbusier 1945, Strukturdiagramm Unité d'habitation

Positionen, Größen und Distanzen haben im topologischen Raum nicht dieselbe Bedeutung wie im kartesischen. In der topologischen Betrachtungsweise reduziert sich alles auf die Verknüpfungen abstrakter Einheiten. Noch bevor ein räumlicher Entwurf existiert, können die Spezifikationen eines Gebäudes in Diagrammen visuell dargestellt werden. Die in Diagrammen offengelegte Organisationsstruktur eines Gebäudes kann im darauffolgenden Entwurfsprozess zu formal unterschiedlichen Ergebnissen führen.

Als Beispiel für eine solche strukturelle Gleichheit (Isomorphie), führt der Architekt William J. Mitchell in seinem Buch *The Logic of Architecture* drei, auf den ersten Blick sehr unterschied-

dliche Entwürfe des Architekten Frank Lloyd Wright an. Obwohl die geometrische Form der in Abbildung 1.5a gegenübergestellten Häuser mit den Namen *Life*, *Jester* und *Sundt* stark variiert, bemerkt Mitchell, dass alle drei Entwürfe dieselben Funktionsbereiche enthalten, und dass die Struktur der Nachbarschaften zwischen diesen Bereichen ident ist.

*»Since room adjacencies are often determined by practical imperatives, but an architect may have considerable freedom to vary shape, we can often find this sort of equivalence among plans for the same type of building.«<sup>19</sup>*

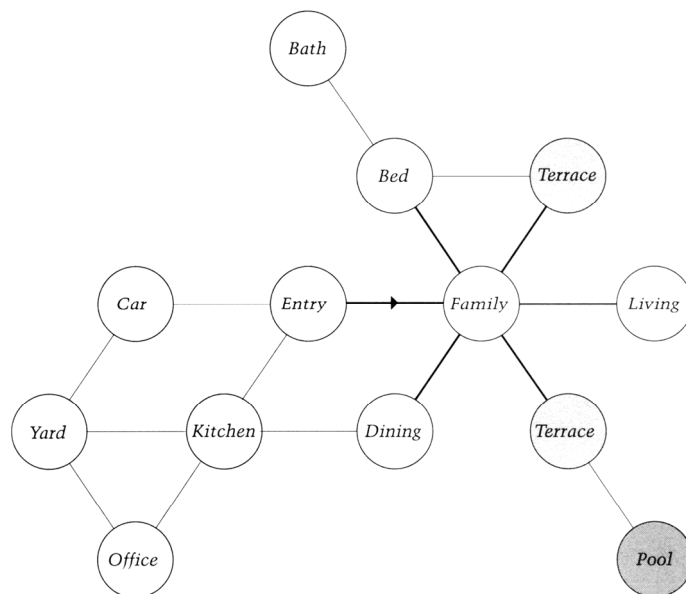


Life House, 1938

Jester House, 1938

Sundt House, 1941

(a)



(b)

Abb 1.5 (a) Wright um 1940, Drei Entwurfslösungen mit gleicher Topologie (b) Mitchell 1990, Strukturdiagramm

<sup>19</sup> Vgl. Mitchell [erstveröffentlicht 1990] 1996, 116ff.

Gilles Deleuze und Felix Guattari bezeichnen Diagramme in ihrem Buch *Tausend Plateaus* als abstrakte *Maschinen*. Nach den Philosophen schränkt eine abstrakte Maschine nicht ein und bestimmt auch nichts vorher - im Gegenteil, sie ermöglicht erst das Veränderliche, Flexible, sich Entwickelnde.<sup>20</sup>

*»Weil eine abstrakte Maschine oder Diagrammatik nicht dazu da ist, um etwas zu repräsentieren, sei es auch etwas Reales, sondern um etwas zukünftig Reales zu konstruieren, einen neuen Typus von Realität. Sie steht also nicht ausserhalb der Geschichte, sondern ist vielmehr der Geschichte immer ‚voraus‘, in jedem Moment, in dem sie Punkte der Schöpfung oder Potentialität konstituiert.«*

## 1.5 Schlussfolgerungen

Architekten erhalten Raumprogramme beispielsweise als Ausschreibungsunterlage bei der Auslobung von Architekturwettbewerben. Selten handelt es sich dabei um exakt definierte Problemstellungen. Die Ziele von Entwurfsaufgaben sind oft vage oder lückenhaft formuliert. Die geäußerten Wünsche stellen sich während des Entwurfsprozesses nicht selten als ambivalent oder gar widersprüchlich heraus. Wie die Ausführungen in diesem Kapitel gezeigt haben, erklären sich die Zielkonflikte unter anderem dadurch, dass ein zukünftiges Gebäude bei der Programmerstellung nur in Fragmenten beschrieben werden kann. Die Gesamtstruktur schwebt zu diesem Zeitpunkt noch im Unbewussten, oft als Konflikt der Kräfte oder Gegensatz der Wünsche<sup>21</sup>, sprichwörtlich zwischen den Zeilen.

Ein Beispiel für eine „unlösbare“ Entwurfsaufgabe skizzierte Jona Friedman (Abb. 1.6). Es könnte in einem Raumprogramm durchaus der Entwurf von vier Räumen gefordert werden, welche einerseits alle direkt miteinander verbunden sind und zugleich jeweils einen Zugang zum Garten aufweisen sollen. Diese Entwurfsaufgabe ist jedoch ohne das Hinzufügen eines weiteren, noch nicht katalogisierten Raums nicht lösbar.<sup>22</sup>

Unpräzise definierte Probleme verlangen zu ihrer Lösung nach einer Ergänzung der Spezifikationen.<sup>23</sup> Viele Eigenschaften und funktionale Anforderungen an zukünftige Gebäude werden in heute üblichen Raumprogrammen nicht explizit beschrieben. Fehlende Informationen werden in der Regel von Architektinnen und Architekten während des Entwurfsprozesses laufend ergänzt. Hierfür ist neben Kreativität auch Routine in allen betroffenen Wissensgebieten gefordert.

---

<sup>20</sup> Vgl. Schmeing/Kleinheinz 1999, 194ff.

<sup>21</sup> Vgl. Deleuze/Guattari [erstveröffentlicht 1980] 1992, 196.

<sup>22</sup> Vgl. Friedman 1975, 39.

<sup>23</sup> Vgl. Franck 2009, 242.



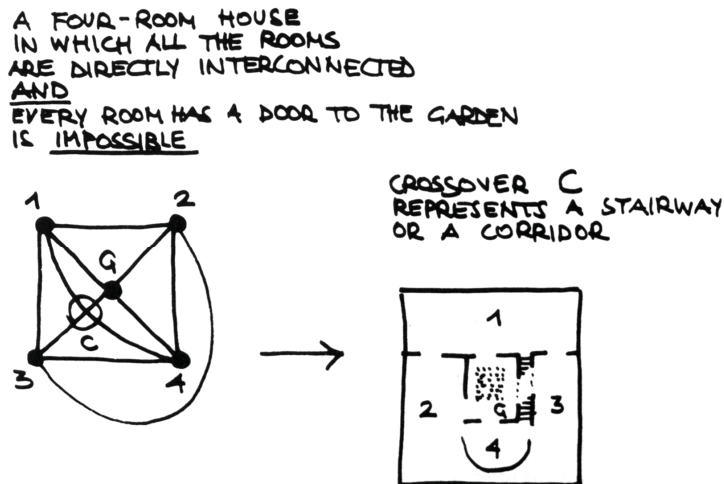


Abb 1.6 Friedman 1975, Unlösbare Entwurfsaufgabe

Unabhängig von der angewandten Entwurfsmethode erfolgt der Entwurf eines Gebäudes in den allermeisten Fällen im Dialog mit den, als Ziele formulierten Anforderungen. Das Raumprogramm kann in diesem Prozess als Bewertungssystem dienen. Es bildet eine Grundlage für die Beurteilung von Entwürfen, welche bei der Selektion von Varianten oder bei Entscheidungen für laufende Abänderungen Anwendung findet. Ein Entwurf erfüllt das Programm, wenn er die geforderten Funktionsbereiche enthält, wenn das Gebäude als Ganzes, die unterschiedlichen Raumgruppen und jeder der Räume und Zonen die gewünschten Merkmale aufweist beziehungsweise, wenn die Räume im Entwurf zueinander so angeordnet sind, dass auch die erforderlichen Beziehungen zwischen ihnen bestehen. Merkmale in einem Raumprogramm vorzuschreiben macht nur dann Sinn, wenn diese in einem Gebäudeentwurf auch eindeutig identifiziert werden können. Nur unter dieser Voraussetzung kann die Abweichung von einem erklärten Ziel festgestellt und für die Beurteilung einer Lösung herangezogen werden. Die Erfüllung eines Programms sagt dabei nicht immer etwas über die architektonische Qualität eines Gebäudeentwurfs aus. Erfüllt ein Entwurf zwar das Programm, das geplante Gebäude aber nicht seinen Zweck, so war das Programm möglicherweise unpassend oder zu unpräzise formuliert.

*»Denn ein Problem hat immer die Lösung, die es nach der Art, in der es gestellt ist, und dem symbolischen Feld, über das man verfügt, um es zu stellen, verdient. [...] Einschließlich der Absurditäten, Schändlichkeiten und Grausamkeiten, welche die ‚Lösung‘ auf Grund der Struktur nach sich ziehen.«<sup>24</sup>*

<sup>24</sup> Vgl. Deleuze [erstveröffentlicht 1969] 1992, 35.

Erfolgt der Entwurf eines Gebäudes im Dialog mit den formulierten Zielen, so fällt bereits dem Programm indirekt ein verantwortungsvoller gestalterischer Aspekt zu.

Nach der Anreicherung mit *implizitem Wissen*<sup>25</sup> der Entwerferin oder des Entwerfers ist das Programm, nach dem Architektur tatsächlich entworfen wird, in den meisten Fällen wesentlich komplexer als beispielsweise das in den Auslobungsunterlagen eines Wettbewerbs enthaltene Briefing. Raumprogramme können durchaus als Messlatte für bestimmte Projektziele betrachtet werden. Als Bewertungssystem sind sie in beschränktem Umfang, beispielsweise bei der Vorprüfung von Wettbewerbsbeiträgen, einsetzbar. Es hat sicher auch seine gute Seite, dass Raumprogramme eine unvollständige Wissensbasis darstellen. Architektinnen und Architekten haben dadurch die Anforderungen an zukünftige Gebäude nicht nur zu erfüllen, sondern werden selbst indirekt in den Prozess der *Programmierung* von Architektur miteinbezogen.

---

<sup>25</sup> Implizites oder stilles Wissen (Tacit Knowledge) bedeutet, vereinfacht ausgedrückt, „können, ohne sagen zu können, wie“. Jemand „weiß, wie es geht“, aber sein Wissen steckt implizit in seinem Können, ihm fehlen die Worte, um dieses Können zu beschreiben oder es anderen verbal zu vermitteln. Wikipedia, dt.: stilles Wissen, 20.10.2014.

## 2 GRAPHEN – LERNEN VON KÖNIGSBERG

*»Neben jenem Bereich der Geometrie, der die Größen untersucht und zu allen Zeiten eifrig studiert wurde, gibt es noch einen anderen, bis jetzt beinahe unbekannt, den Leibniz als Erster erwähnt und Geometrie der Lage (Geometriam situs) genannt hat. Gegenstand der Untersuchung ist das, was nur durch die Lage bestimmt werden kann und die Ergründung der Eigenschaften dieser Lage; hierbei sollen die Größen außer Acht gelassen und das Rechnen mit Größen nicht angewendet werden.«<sup>1</sup>*

Leonhard Euler 1736

Mit diesen Worten begann Leonhard Eulers Aufsatz zur Lösung eines scheinbar trivialen Alltagsproblems der Bewohnerinnen und Bewohner einer kleinen ostpreußischen Stadt in der ersten Hälfte des 18. Jahrhunderts (Abb 2.1). Heute gilt das Dokument als erste topologische und graphentheoretische Abhandlung der Geschichte. Das folgende Kapitel vollzieht einen Zeitsprung zurück ins frühe 18. Jahrhundert, der Geburtsstunde dieses verhältnismäßig jungen Teilgebiets der Mathematik. Nach einer darauffolgenden kurzen Einführung in die Grundsätze der Graphentheorie widmet sich das Kapitel dem Einsatz von Graphen im Bereich der Architektur und benachbarter Wissenschaftsgebiete. Ohne den Anspruch auf Vollständigkeit zu erheben, soll sich die beispielhafte Auswahl der graphentheoretischen Anwendungen dabei nur auf Themen beschränken, welche zum inhaltlichen Verständnis der vorliegenden Arbeit beitragen können.

### 2.1 Das Königsberger Brückenproblem

Es heißt, dass im frühen 18. Jahrhundert die Bewohnerinnen und Bewohner Königsbergs ein Rätsel unterhielt: Die Stadt war vom Fluss Pregel in unterschiedliche Landabschnitte geteilt, welche durch sieben Brücken verbunden waren. Unter den leidenschaftlichen Flaneuren kursierte die Frage, ob es einen Weg durch alle Stadtteile gab, der jede Brücke genau einmal überquerte. Diese viel diskutierte Frage wurde im Jahre 1736 an den Schweizer Mathematiker Leonhard Euler herangetragen und sollte als das „Königsberger Brückenproblem“ in die Geschichte der Mathematik eingehen. Euler hielt das Problem für banal. Die Lösung beruhe auf reiner Argumentation und bedürfe laut seinen eigenen Ausführungen weder der Geometrie noch der Algebra noch der Kunst der Kombinatorik.<sup>2</sup> Dennoch widmete er der Aufgabe seine Aufmerksamkeit. Noch im selben Jahr konnte der Mathematiker den gewünschten Beweis liefern. In seiner Schrift *Solutio problematis ad geometriam situs pertinentis* zeigte Euler, dass in Königsberg kein Weg mit den beschriebenen Eigenschaften existiert.

---

<sup>1</sup> Euler 1736 *Solutio*, Übersetzung aus dem Lateinischen, in: Velminski 2009, 13.

<sup>2</sup> Vgl. Euler 1936 Marinoni.



Abb 2.1 Matthäus Merian um 1640: Kupferstich (38X29cm), Königsberg in der Vogelschau

Euler ordnete das Problem der *Geometrie der Lage* (*geometria situs*) zu, welche von Gottfried Wilhelm Leibniz und später auch von Christian Wolff propagiert wurde.<sup>3</sup> Zur Lösung des Königsberger Brückenproblems entwickelte Euler ein Verfahren, welches im Grunde aus zwei Abstraktionsschritten bestand.

Um die Konzentration auf das Wesentliche zu lenken, ersetzte er zuerst die Landkarte der Stadt Königsberg durch ein simples Diagramm, welches nur die Elemente enthielt, welche für den gewünschten Nachweis notwendig waren (Abb 2.2). Kartografische Details wie Straßen, Bauwerke etc. wurden weggelassen. Die exakte Figur der Stadtteile, Flusskrümmungen, Brückenlängen und Distanzen zwischen den Brücken waren für die Lösung des Problems nicht von Bedeutung. Danach begann Euler mit der Codierung der verbliebenen Elemente. Die Landabschnitte, gleich ob Insel oder Festland, wurden durch die Großbuchstaben A, B, C und D bezeichnet. Die sieben Brücken erhielten die Kleinbuchstaben a, b, c, d, e, f und g. In der Folge galt Eulers Interesse nur noch den Buchstaben. Aus der Stadt war eine Verkettung von Symbolen geworden. Auf diese Weise konnte jeder Weg durch Königsberg durch eine Zeichenkette ausgedrückt werden.

<sup>3</sup> Vgl. Velminsky 2009, 160.

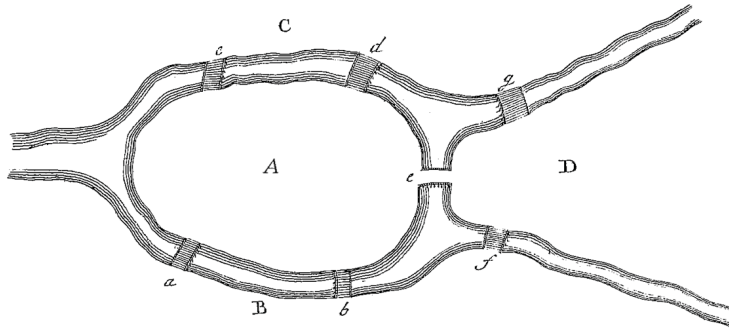


Abb 2.2 Euler 1736: Diagramm der Stadt Königsberg

»Wenn also einer vom Gebiet A in das Gebiet B gelangt über die Brücke a oder b, so bezeichne ich diesen Übergang mit den Buchstaben AB, deren erster das Gebiet angibt, aus welchem der Wanderer herauskommt, während der zweite das Gebiet angibt, in das er nach Überschreitung der Brücke gelangt. Wenn der Wanderer darauf aus dem Gebiet B über die Brücke f in das Gebiet D geht, so wird dieser Übergang mit den Buchstaben BD bezeichnet; diese beiden hintereinander ausgeführten Übergänge AB und BD bezeichne ich nun bloß mit den drei Buchstaben ABD, weil der mittlere, B, sowohl das Gebiet angibt, in das der erste Übergang hineinführt, als auch das Gebiet, aus dem der zweite Übergang herausführt.«<sup>4</sup>

Ein Spaziergang über zwei Brücken kann nach diesem System durch drei Zeichen buchstabiert werden, über drei Brücken durch vier, über vier Brücken durch fünf Zeichen oder allgemein ausgedrückt, durch eine um eins größere Zahl von Buchstaben als überschrittene Brücken. Ein Rundgang über alle sieben Brücken der so kodierten Stadt Königsberg ergibt also in jedem Fall eine Reihe von acht Buchstaben. Euler war der Lösung damit bereits ein Stück näher gekommen. Als nächstes interessierte sich Euler für die Häufigkeit jedes der vier Großbuchstaben innerhalb dieser Reihe.

»Wenn der Wanderer diese Brücke [a] überschreitet, so muss er sich vor dem Übergang entweder in A befunden haben oder er gelangt nach dem Übergang nach A; in der obigen Bezeichnungsweise wird also der Buchstabe A gerade einmal auftreten. Falls drei Brücken a, b, c nach A führen und der Wanderer alle drei überschreitet, wird der Buchstabe A zweimal in der Wegbezeichnung vorkommen, gleichgültig ob der Weg in A begonnen hat oder nicht. Und wenn fünf Brücken nach A führen, so wird der Buchstabe A dreimal in der Bezeichnung des Weges auftreten, der über diese fünf Brücken führt.«<sup>5</sup>

<sup>4</sup> Euler 1736 Solutio, Übersetzung aus dem Lateinischen, in: Velminski 2009, 14f.

<sup>5</sup> Ebd. 16.

Der Mathematiker hatte erfasst, dass die Häufigkeit jedes Großbuchstabens von der Anzahl der Brücken, die zu ihm führen, abhängt. Er stellte weiters fest, dass es dabei auch von Bedeutung ist, ob diese Anzahl gerade oder ungerade ist. Bei einer geraden Zahl von Brücken schreibt Euler als Häufigkeit die Hälfte der Zahl vor. Bei ungerader Zahl von Brücken beträgt die Häufigkeit die Hälfte der um eins vergrößerten Zahl. Die ganze Fragestellung hatte sich darauf reduziert „ob aus den vier Buchstaben A, B, C und D eine Reihe von acht Buchstaben gebildet werden kann, in der alle diese Folgen in der vorgeschriebenen Anzahl auftreten“<sup>6</sup>. Eulers allgemeinen Untersuchungen zufolge muss der Buchstabe A dreimal in dieser Reihe vorkommen, da fünf Brücken nach A führen. B, C und D müssen dagegen nur zweimal vorkommen, da nur jeweils drei Brücken in die durch diese Buchstaben bezeichneten Gebiete führen.  $3+2+2+2$  ergibt 9. Neun Buchstaben in einer Reihe von acht Buchstaben anzuordnen war unmöglich.

*»[D]araus ist ersichtlich, dass der gesuchte Übergang über die sieben Königsberger Brücken nicht ausgeführt werden kann.«<sup>7</sup>*

Der Mathematiker Euler hatte das Königsberger Brückenproblem gelöst, war aber daran interessiert, auch in beliebigen anderen topografischen Konstellationen mühelos entscheiden zu können, ob eine solche Anordnung der Buchstaben möglich ist. Aus diesem Grunde formulierte Euler die nötigen Handlungsschritte allgemein:

*»Zunächst bezeichne ich die einzelnen Gebiete, die durch das Wasser voneinander getrennt sind, mit Buchstaben A, B, C usw.*

*Zweitens nehme ich die Zahl aller Brücken, vermehre sie um eins und schreibe die so entstehende Zahl zuoberst auf.*

*Darunter schreibe ich drittens die Buchstaben A, B, C usw. und neben jeden derselben die Zahl der Brücken, die zu seinem Gebiet führen.*

*Viertens versehe ich diejenigen Buchstaben, neben denen gerade Zahlen stehen, mit einem Stern.*

*Fünftens notiere ich neben diesen geraden Zahlen ihre halbe Summe, neben die ungeraden aber schreibe ich die Hälfte der um eins größeren Zahl.*

*Sechstens addiere ich diese zuletzt erhaltenen Zahlen. Wenn diese Summe um eins kleiner oder gleich der oben als Erste festgehaltenen Zahl ist, dann schließe ich, dass der gewünschte Übergang ausgeführt werden kann. Indes muss man beachten: Wenn die Summe um eins kleiner ist als die oben stehende Zahl, dann muss man den Spaziergang in einer Gegend beginnen, die mit einem Stern versehen ist. Im anderen Fall, wenn die beiden Zahlen also gleich sind, muss man in einem Gebiet ohne Stern beginnen.«<sup>8</sup>*

---

<sup>6</sup> Ebd. 17.

<sup>7</sup> Ebd.

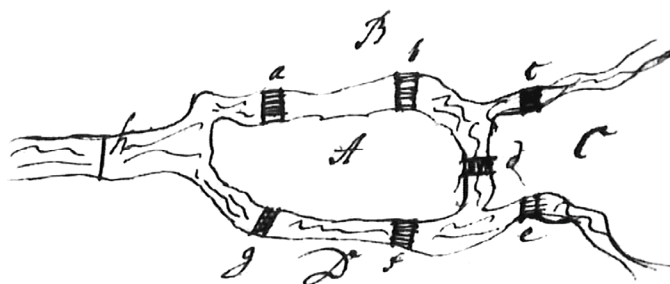
<sup>8</sup> Ebd. 18f.

»Nachdem man herausgefunden hat, ob ein Weg existiert, bleibt noch die Frage, wie man ihn führen muss. Hierzu dient folgende Regel: Man lasse in Gedanken, so oft das geht, zwei Brücken, die dieselben zwei Gebiete verbinden, weg, wodurch die Zahl der Brücken meistens außerordentlich vermindert wird. Dann suche man, was leicht ist, einen Weg der gewünschten Art über die übrig Bleibenden. Hat man ihn gefunden, so werden die Brücken, die man in Gedanken weggelassen hat, den Weg nicht mehr wesentlich stören, wie man nach kurzem Überlegen leicht sieht; ich glaube daher nicht, dass es nötig sein wird, weitere Erklärungen zur Ermittlung der Wege vorzubringen.«<sup>9</sup>

		8
A	5	3
B	3	2
C	3	2
D	3	2
		9

Abb 2.3 Rechnung

Euler hatte das Königsberger Brückenproblem nicht nur theoretisch gelöst, in einem Brief an den italienischen Mathematiker Giovanni Jacopo Marinoni schlägt Euler am 3. März 1736 die Errichtung einer achten, mit „h“ bezeichneten Brücke zwischen den diesmal mit den Buchstaben B und D benannten Gebieten der Stadt Königsberg vor (Abb 2.4a). Abbildung 2.4b zeigt die erneute Berechnung laut Eulers Anweisungen. Die Rechnung (Abb 2.4b) gibt Eulers Entwurfsskizze Recht. Der ersehnte Rundgang durch die Stadt kann unter Einbezug der neuen Brücke ausgehend von Gebiet A oder C als Startpunkt, ausgeführt werden. Der Vorschlag Eulers, eine achte Brücke zu errichten, wurde erst im Jahre 1905 in die Realität umgesetzt.<sup>10</sup>



(a)

		9
A	5	3
B*	4	2
C	3	2
D*	4	2
		9

(b)

Abb 2.4 (a) Euler 1736: Skizze mit dem Vorschlag einer achten Brücke „h“. (b) Rechnung

<sup>9</sup> Ebd. 22f.

<sup>10</sup> Vgl. Velminsky 2009, 168.

## 2.2 Topologie und Graphentheorie

Aus heutiger Sicht beschäftigte die Bewohnerinnen und Bewohner von Königsberg ein topologisches Problem, zu dessen Lösung Euler Methoden entwickelte, welche der Graphentheorie zugerechnet werden.<sup>11</sup> Die auf Leibniz zurückgehende *geometria situs* etablierte sich gegen Ende des 19. Jahrhunderts als eigenständige Disziplin unter dem heute gebräuchlichen Namen Topologie. In diesem Teilgebiet der Mathematik beschäftigt man sich mit den Eigenschaften mathematischer Strukturen, welche unter stetigen Verformungen - Dehnung, Stauchung oder Verzerrung - erhalten bleiben. Das grundlegende Konzept der Topologie ist die Nachbarschaft von Orten. Dabei werden metrische Abstände abstrahiert. Wenn sich zwei Strukturen durch Deformation ineinander überführen lassen, so werden diese als topologisch gleich oder äquivalent bezeichnet; sie sind hingegen topologisch unterschiedlich, wenn dafür diskontinuierliche Operationen wie Reißen oder Kleben angewandt werden müssen.

Die Graphentheorie stellt wiederum ein Teilgebiet der Topologie dar. Ein Graph ist eine abstrakte Struktur aus einer endlichen Menge von Knoten und Kanten, die eine Menge von Objekten mit den zwischen diesen Objekten bestehenden Verbindungen repräsentiert. Üblicherweise werden die Knoten grafisch durch dickere Punkte visualisiert. Die Kanten sind Verbindungselemente zwischen jeweils zwei Knoten und werden durch Geraden oder Bögen dargestellt. Zwei Knoten können auch mehrfach durch Kanten verbunden sein. Werden zwei Knoten durch mindestens eine Kante verbunden, so nennt man diese Knoten *benachbart*. Knoten ohne Nachbarn werden *isolierte Knoten* genannt. Die Anzahl der Kanten, die von einem Knoten ausgehen, wird als dessen *Grad* bezeichnet. Eine Folge aneinanderstoßender Kanten wird als *Zyklus* bezeichnet. Nicht zusammenhängende Graphen zerfallen in sogenannte Teilgraphen.

Der Fluss Pregel, der Königsberg in die von Euler mit Großbuchstaben bezeichneten Gebiete teilt, kann abstrakt als Graph dargestellt werden (Abb 2.5). Die verschiedenen Flussarme bildet die Grenze zwischen den vier Gebieten A, B, C und D. Die einzelnen Teilstrecken des Flusses bilden die Kanten der Graphenstruktur, die Flussmündungen können als Knoten des Graphen aufgefasst werden.

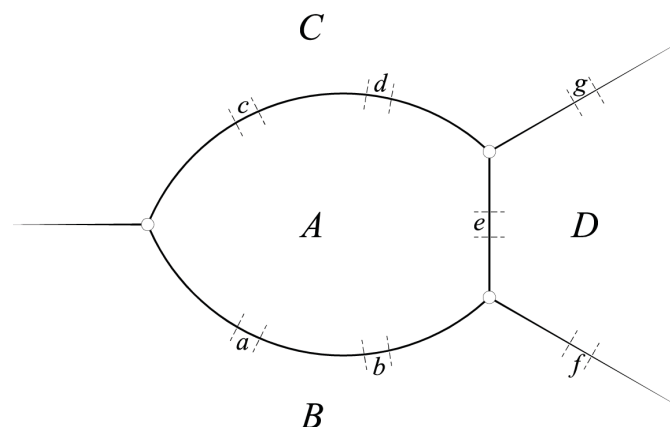


Abb 2.5 Stadt Königsberg als Graph

<sup>11</sup> Vgl. Biggs 1998, 9.



In den folgenden Unterkapiteln sollen einige, für das Verständnis der vorliegenden Arbeit wesentliche Aspekte der Graphentheorie erläutert werden. Anhand des Beispiels der Stadt Königsberg werden in der Folge die Begriffe *Dualer Graph*, *Isomorphie*, *Adjazenzmatrix* und *Gewichtung* eingeführt.

## Dualer Graph

Ein Graph  $G$  ist im mathematischen Sinn ein Tupel, aus einer Menge von Knoten  $V$  und Kanten  $E$ . Der duale Graph  $G'=(V',E')$  eines Graphen  $G=(V,E)$  entsteht, indem in jeder (durch Kanten begrenzten) Fläche des Graphen  $G$  ein neuer Knoten  $v'$  hinzugefügt wird und für jede Kante in  $G$  eine neue Kante  $e'$  erstellt wird, welche die Knoten  $v'$  der beiden angrenzenden Flächen verbindet.<sup>12</sup> Die Bezeichnung *dual* wird benutzt, da diese Eigenschaften symmetrisch gelten: Ist  $G'$  der duale Graph von  $G$ , dann ist auch  $G$  der duale Graph von  $G'$ .<sup>13</sup> In Abbildung 2.6 wird das Wegenetz der Stadt Königsberg abstrakt in Form eines Graphen dargestellt. Dieser entspricht im Grunde Eulers topologischem Denkmodell. Abgesehen davon, dass der Knoten  $A$  entsprechend der realen Anzahl von Brücken ( $a,b,c$  und  $d$ ) durch nicht nur eine, sondern jeweils zwei Kanten mit den Knoten  $B$  und  $C$  verbunden wurde, handelt es sich hierbei im Prinzip um den dualen Graphen der Gebiete und Flussläufe aus Abbildung 2.5. Eulers Suche nach einer achtstelligen Buchstabensequenz entspricht in der Graphentheorie der Suche nach einem *Zyklus* innerhalb dieses Graphen, der die besondere Eigenschaft besitzt, jede Kante genau einmal zu nutzen. Ein solcher *Zyklus* wird heute nach seinem Entdecker als *Eulertour* bezeichnet und ein Graph, der einen solchen *Zyklus* enthält, als *eulerscher Graph*.<sup>14</sup>

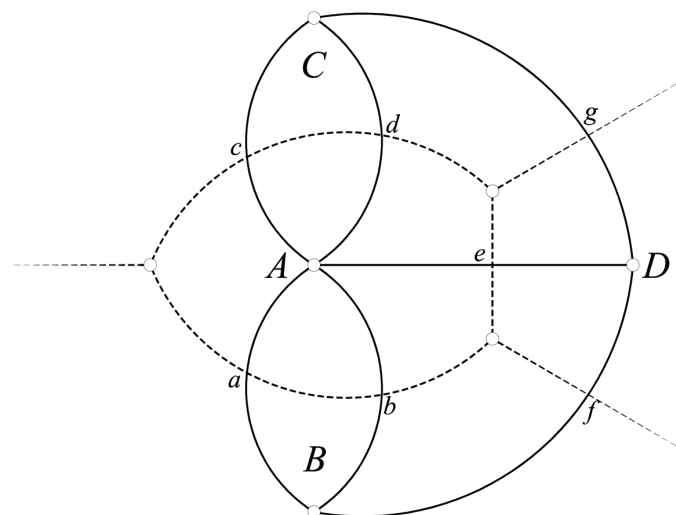


Abb 2.6 Graph und dualer Graph

<sup>12</sup> Voraussetzung hierfür ist, dass es sich um einen *zusammenhängenden* und *planaren* Graphen handelt. Das heißt, je zwei Kanten müssen durch eine Kantenfolge des Graphen verbunden werden können, und der Graph muss auf einer Ebene dargestellt werden können, ohne dass sich Kanten schneiden.

<sup>13</sup> Vgl. Wikipedia, dt., Dualität (Mathematik), 20.10.2014.

<sup>14</sup> Vgl. Biggs 1998, 9.

## Isomorphie

Bei Graphen kommt es einzig und allein darauf an, welche Knoten durch eine Kante miteinander verbunden sind. Dabei ist weder Form oder die Länge der Kanten noch die exakte Lage der Knoten von Bedeutung. Graphen sind strukturell gleich (isomorph), wenn diese unabhängig von ihrer geometrischen Darstellungsform die gleichen Knoten und Kanten aufweisen. Solche bedeutungsgleiche Graphen werden als *isomorph* bezeichnet. Die Struktur eines Graphen bleibt unter stetiger Verformung (siehe Seite 36) erhalten.<sup>15</sup>

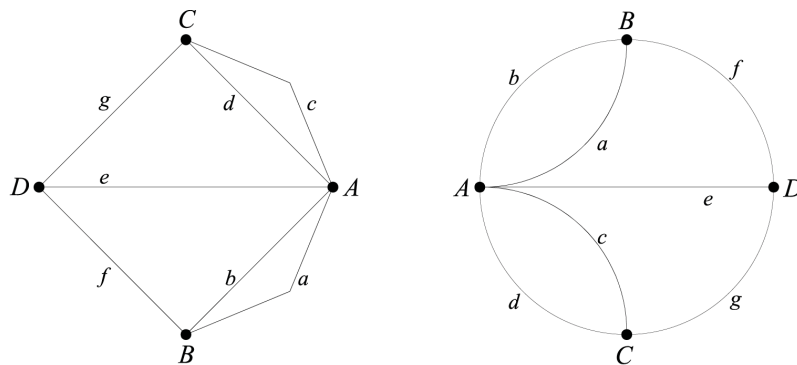


Abb 2.7 Isomorphe Graphen

## Adjazenzmatrix

Jede Graphenstruktur kann auch als Adjazenzmatrix dargestellt werden (Abb 2.8). Jeder Matrix-Eintrag beschreibt eine Kantenbeziehung zwischen zwei Knoten. Dabei steht der Eintrag 1 für eine Kante, der Wert 0 hingegen steht für keine Kante. Bei ungerichteten Graphen, wie dem der Stadt Königsberg, handelt es sich im mathematischen Sinn um symmetrische Relationen. Die Menge aller Beziehungen bildet eine symmetrische, das heißt quadratische, an der Hauptdiagonale gespiegelte Matrix. Die Beziehungen entlang der Hauptdiagonale sind selbstreferenziell (A-A, B-B, etc.) und werden daher von den Betrachtungen ausgeklammert. Die Repräsentation eines Graphen als Matrix erlaubt den Einsatz von Methoden der linearen Algebra<sup>16</sup>

$$\begin{array}{c}
 A \quad B \quad C \quad D \\
 \begin{array}{l}
 A \\
 B \\
 C \\
 D
 \end{array}
 \begin{pmatrix}
 - & 1 & 1 & 1 \\
 1 & - & 0 & 1 \\
 1 & 0 & - & 1 \\
 1 & 1 & 1 & -
 \end{pmatrix}
 \end{array}$$

Abb 2.8 Graph in Matrixform

<sup>15</sup> Encyclopedia of Mathematics: Graph Isomorphism, 25.10.2014.

<sup>16</sup> Vgl. Wikipedia dt., Adjazenzmatrix, 17.05.2014.

## Gewichtung

Sollen Graphen mit differenzierteren Informationen versehen werden, welche über die reinen Knotenbeziehungen hinausgehen, können die Knoten und Kanten eines Graphen durch das Anhängen von Daten erweitert werden. Dabei wird das Tupel  $(V, E)$  beispielsweise zu einem Tripel  $(V, E, f)$  ergänzt. Graphen, deren Knoten oder Kanten auf diese Weise mit zusätzlichen Informationen versehen sind, nennt man gewichtet. Handelt es sich dabei um natürliche Zahlen, so spricht man von Färbung. Werden den Knoten oder Kanten eines Graphen Namen beigelegt, spricht man von benannten Graphen.<sup>17</sup> Ein Graph kann zugleich mehrere Erweiterungen beinhalten. Ein anschauliches Beispiel stellt die heute international gebräuchliche Darstellung von Liniennetzplänen öffentlicher Verkehrsmittel in Graphenform dar.

## 2.3 Liniennetze

Bis zum Anfang des 20. Jahrhunderts war es üblich, Liniennetzpläne der Bahn oder U-Bahn auf der Basis geometrisch korrekter Stadtpläne oder Landkarten topografisch „richtig“ darzustellen. In der Folge lagen Bahnstationen im Zentrum einer Stadt häufig sehr nahe beisammen. Stationen in Vororten lagen im Gegensatz dazu aufgrund der längeren Verbindungsstrecken sehr weit auseinander. Der britische technische Zeichner und Grafikdesigner Harry Beck erkannte anfang des 20. Jahrhunderts, dass die geografische Genauigkeit der Fahrpläne für die Fahrgäste aber gar nicht von Bedeutung war. Beck vertrat den Standpunkt, dass die Fahrgäste nur wissen wollten, wie man von einer Station zu einer anderen gelangen und wo man von der einen Linie in eine andere umsteigen könne. In anderen Worten: Nur die Topologie, also die räumlichen Beziehungen der Stationen untereinander war entscheidend.

Dieser Erkenntnis folgend entwarf Beck den ersten Plan der Londoner U-Bahn in Graphenform. Die Abbildungen 2.9 und 2.10 zeigen die Gegenüberstellung eines geografisch unverzerrten Plans der London Underground aus dem Jahre 1908 und Becks Entwurf desselben Liniennetzplans aus dem Jahre 1933. Beide Pläne sind, abgesehen von neu entstandenen Stationen, strukturell gleich (isomorph). Die Darstellung von Liniennetzplänen als Graphen stellt heute einen internationalen Standard dar und markiert einen radikalen Durchbruch im Informationsdesign.<sup>18</sup> Die Haltestellen bilden die Knoten, Gleisverbindungen zwischen zwei Haltestellen die Kanten des Graphen. Die unterschiedlichen U-Bahnlinien werden üblicherweise durch unterschiedliche Farben dargestellt. Die Knoten tragen die Namen der Haltestellen. U-Bahn Pläne stimmen seit Beck geometrisch nicht mit dem Stadtplan überein. Die Längen der Kanten entsprechen nicht den überwundenen Distanzen zwischen den Stationen und die Positionen der Knotenpunkte nicht den geografischen Positionen der Haltestellen in der Stadt. Beim Layout solcher Pläne geht es vor allem um eine möglichst übersichtliche Entflechtung und Linienführung. Informationen über die wahren Distanzen zwischen den einzelnen Haltestellen können über numerische Angaben der einzelnen Streckenlängen als Gewichte in den U-Bahn-Graphen einfließen.

---

<sup>17</sup> Vgl. Wikipedia dt., Graph(Graphentheorie), 25.10.2014.

<sup>18</sup> Vgl. Dobbin 2012, 17.

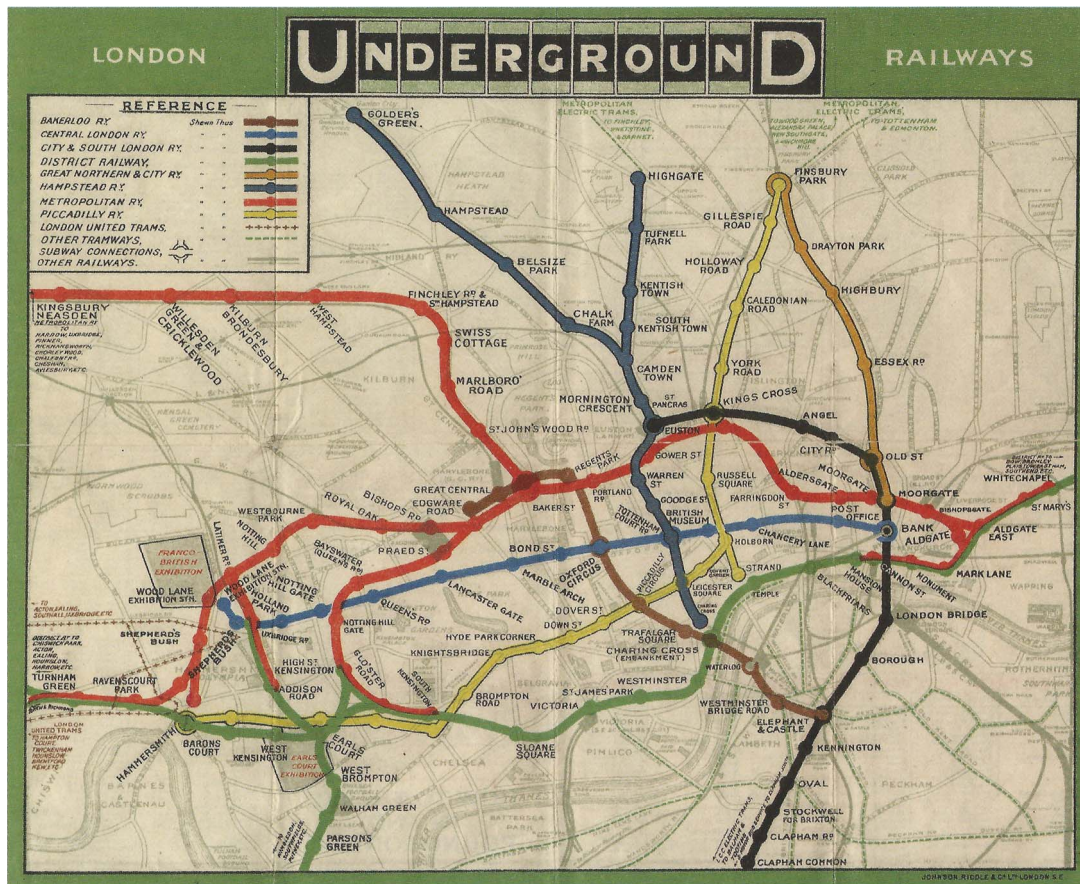


Abb 2.9 Unbekannter Künstler 1908: Londoner U-Bahn Netz



Abb 2.10 Harry Beck 1933, Londoner U-Bahn Netz

Abbildung 2.11 zeigt einen Graphen, welcher die Flugrouten einer tschechoslowakischen Flug-Transportgesellschaft aus dem Jahre 1933 darstellt. Der Informationswissenschaftler und Grafikdesigner Edward Tufte hat die besonderen Qualitäten dieser Informationsgrafik in seinem Buch *Envisioning Information, Narratives of Space and Time* aus dem Jahre 1990 hervorgehoben:

*»A comprehensive narrative discription of a transport system requires a record of both time and spatial experiences. Here a complex network of routes is brought together with flight times and identification numbers in a brilliant map/schedule für the Czechoslovakia Air Transport Company in 1933.«<sup>19</sup>*

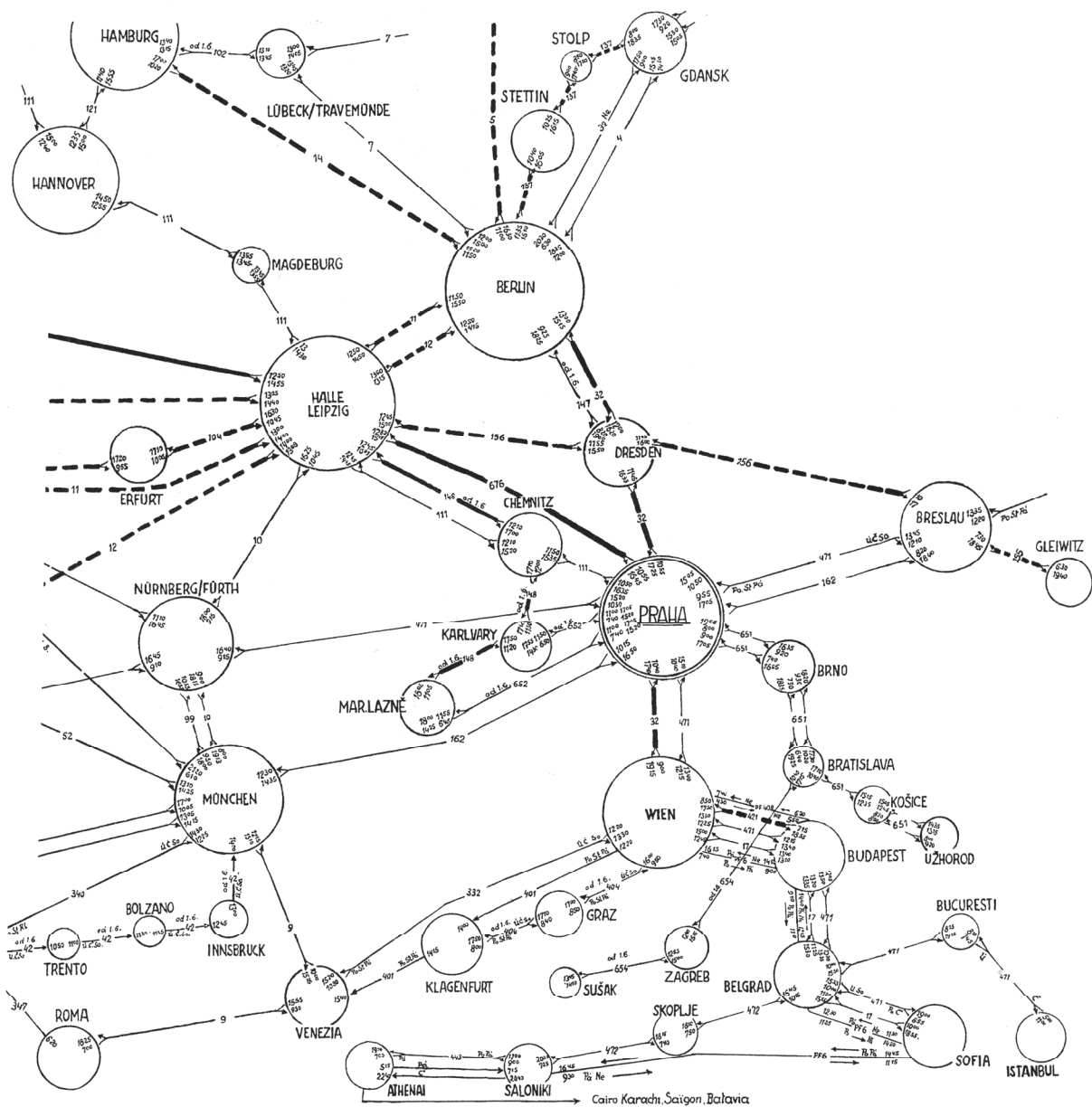


Abb 2.11 Czechoslovakia Air Transport Company 1933, Flugplan

<sup>19</sup> Tufte 1990, 102.

Der in Abbildung 2.11 dargestellte Graph des Flugroutennetzes ist mehrfach gewichtet. Städte bilden die Knoten des Graphen und werden als Kreise dargestellt, deren unterschiedliche Durchmesser die Größe der Stadt oder die Anzahl der Flugverbindungen repräsentieren. Jeder Knoten ist mit dem Namen der Stadt benannt, die er verkörpert. Die Kanten werden durch jeweils einen Hin- und Rückflug gebildet. Die meisten Städte sind durch parallele Mehrfachkanten miteinander verbunden. Jede Kante ist mit einer Flugnummer gewichtet. Weitere Gewichtungen erfolgen durch verschiedene Strichstärken und Linientypen. An den Enden der einzelnen Kanten kann die genaue Abflug- und Ankunftszeit abgelesen werden.

## 2.4 Graphen in der Architektur

Die Einsatzmöglichkeiten von Graphen in der Architektur wurden in der zweiten Hälfte des 20. Jahrhunderts intensiv erforscht. Graphen wurden häufig dazu benutzt, um morphologisch-strukturelle Eigenschaften von Gebäuden, wie beispielsweise Nachbarschaft, Nähe, Zentralität, Tiefe etc., unabhängig von der geometrischen Form auf einer abstrakten topologischen Ebene zu untersuchen oder miteinander zu vergleichen. Die Arbeiten von J.P. Steadman (Abb 2.13), Yona Friedman, Julienne Hanson und Bill Hillier (Space Syntax) oder die Raumanalysen des Quickborner Teams stellen bekannte Beispiele methodisch verwandter Arbeitsweisen dar.<sup>20</sup>

*»[I]n order to compare dwellings with one another and to interpret their sociological significance, we have to solve a prior problem, that of identifying the elements and relations which make up the space pattern. To compare spatial patterns we have to know what a pattern is, and how to tell one configuration from another. Configuration, in this instance, means something quite precise. Spatial relations exist where there is any type of link between two spaces. Configuration exists when the relations which exist between two spaces are changed according to how we relate each to a third, or indeed to any number of spaces. Configurational descriptions therefore deal with the way in which a system of spaces is related together to form a pattern, rather than with the more localised properties of any particular space.«<sup>21</sup>*

Ähnlich wie die Stadt Königsberg kann auch die Struktur eines Gebäudes abstrakt als Graph dargestellt werden. Analog zu den Flussläufen stellen dabei Wände und Decken die wesentlichen trennenden Elemente dar, welche Raum beziehungsweise das Bauvolumen in Teilräume gliedern. Die einzelnen Räume werden wiederum durch Wandöffnungen oder Türen kontrolliert miteinander verbunden. Von Gebäuden können unterschiedliche Graphen abgeleitet werden. Die geometrische Form des Gebäudes tritt danach in den Hintergrund und wird für die weiteren Betrachtungen nicht mehr benötigt.

<sup>20</sup> Vgl. Friedman 1775, Steadman 1983, Hillier 1996, Hillier/Hanson 1984, Hanson 1998.

<sup>21</sup> Hanson 1998, 23.

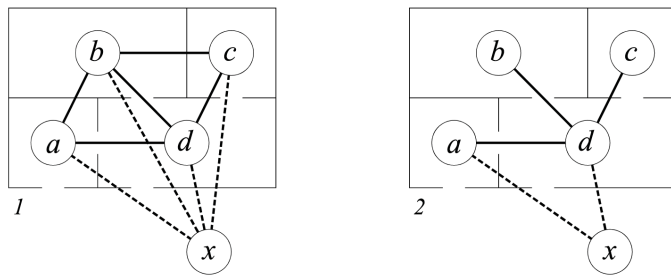


Abb 2.12 Adjacency- und Access Graph desselben Grundrisses

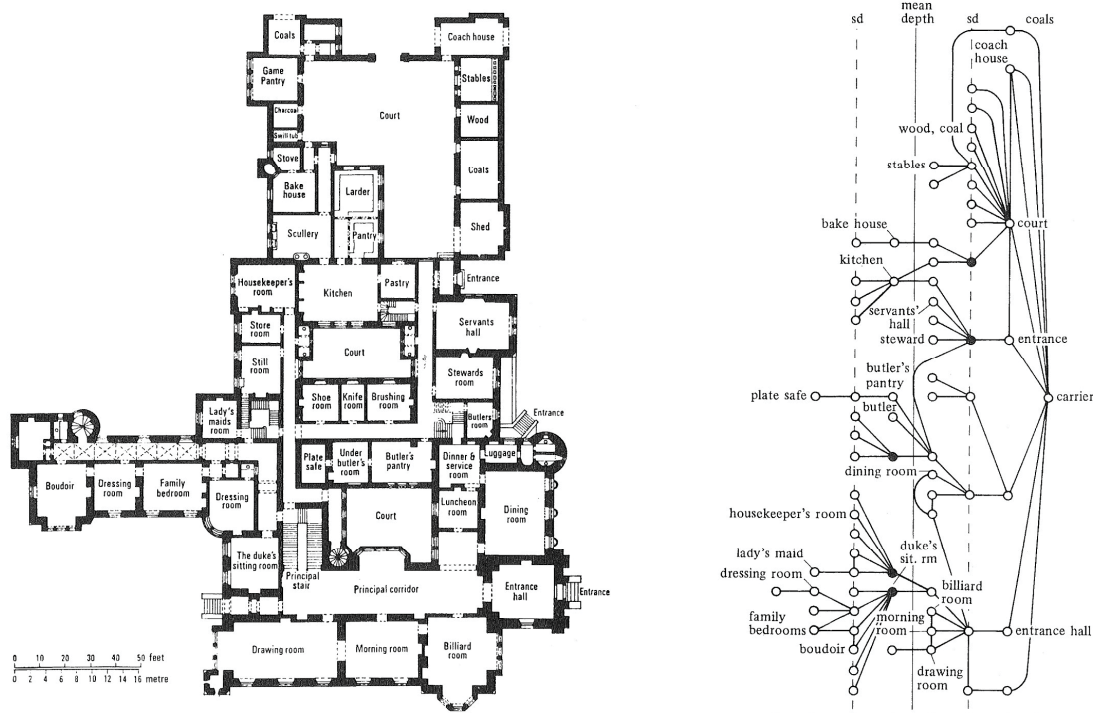


Abb 2.13 Steadman 1983, Grundriss und (justified) Access Graph eines Gebäudes

Die einzelnen Räume werden dabei in der Regel als Knoten aufgefasst. Werden jene Knoten durch Kanten miteinander verbunden, welche geometrisch benachbarte Räume repräsentieren, so erhält man den Graphen der Nachbarschaften (Abb 2.12.1 Adjacency Graph). Der Graph der räumlichen Zugänglichkeit (Abb 2.12.2 Access Graph) entsteht hingegen, wenn nur die Knoten jener Räume durch Kanten verbunden werden, welche im Grundriss durch eine Tür verbunden sind. Der das Gebäude umgebende Außenraum kann analog zu den Innenräumen als ein Knoten des Graphen aufgefasst werden (in Abb 2.12 mit X gekennzeichnet). Auch im Forschungsbereich des *Human Engineering*<sup>22</sup> macht man von der Abstraktion von Bauwerken als Graph Gebrauch. Graphen werden hier in Innenräumen zur Analyse „molekularer“ Beziehungen räumlicher Systeme benutzt. Um die Wege des Nutzers eines Gebäudes oder Bedieners einer Maschine zu ermitteln, entwickelten Kantowitz und

<sup>22</sup> Beim *Human Engineering* befasst man sich mit der Analyse menschlicher Bedürfnisse und Fähigkeiten, um Tätigkeit am Arbeitsplatz zu planen oder um angemessene Räume, Abläufe, Möbel etc. zu entwerfen und so die Leistungen der Mitarbeiter zu steigern. Vgl. Wirtschaftslexikon, [www.onpulson.de](http://www.onpulson.de), human engineering, 2.11.2014.

Sorkin 1983 die Methode der *Verknüpfungsanalyse*. Dabei werden die an einer Operation oder an einem Arbeitsprozess beteiligten Elemente wie Einrichtungsgegenstände, Maschinen etc. durch Knoten eines Graphen dargestellt. Beziehungen zwischen diesen Elementen und dem Benutzer werden durch Kanten (Linien) markiert. Danach wird die Wichtigkeit und Häufigkeit jeder Verbindung beurteilt, welche durch unterschiedliche Linienstärken oder in Form von Zahlen dargestellt werden können. Ziel der Methode ist die Reduktion von zu langen oder zu komplexen Interaktionsbeziehungen zwischen Menschen und den Systembestandteilen.<sup>23</sup> Nach diesem Prinzip wurden beispielsweise vom Quickborner Team<sup>24</sup> Verknüpfungen und Informationsflüsse in Großraumbüros mit Hilfe von Graphen als Abstraktion der Funktionsbereiche und Arbeitsplätze analysiert (Abb 2.14-15). Der Grundriss wurde dazu in polygonale Zonen zerteilt, welche Gruppen beziehungsweise Betriebsabteilungen repräsentieren. Beziehungen dieser Funktionsbereiche werden durch Pfeile markiert. Innerhalb der Gruppen erfolgt die Abstraktion der Arbeitsplätze beziehungsweise Tischgruppen in Form von Teilgraphen. Zur eindeutigen Identifikation wurde jeder Knoten mit einer Nummer versehen. Die Stärke oder Wichtigkeit der Beziehungen der unterschiedlichen Bereiche innerhalb einer Gruppe werden durch die Linienstärke der Kanten des Graphen ausgedrückt. Dünne Linien stehen dabei für schwache, dickere für starke Abhängigkeiten.

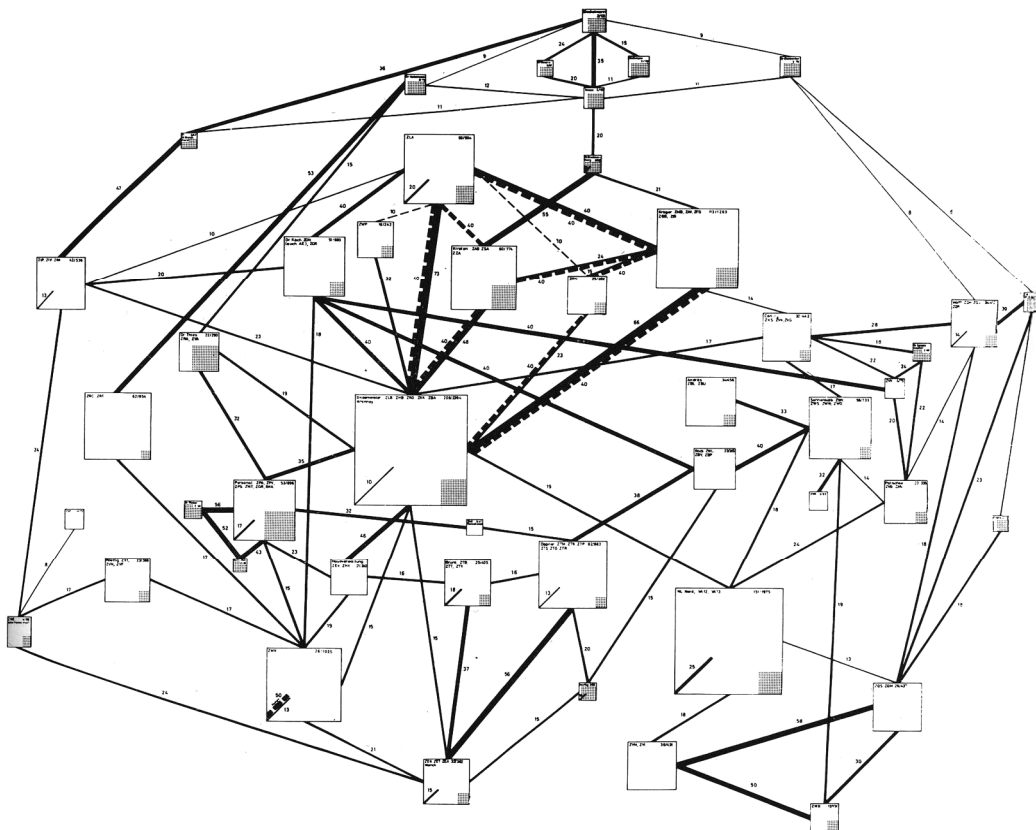


Abb 2.14 Quickborner Team um 1970, Informationsfluss in einem Großraumbüro

<sup>23</sup> Vgl. Kannheiser 1989, 75.

<sup>24</sup> Dem von Eberhard und Wolfgang Schnelle zusammen mit Hermann Dunst im Jahre 1956 gegründete, heute international tätige Beratungsunternehmen ([www.quickborner-team.de](http://www.quickborner-team.de)) für weite Bereiche der Gebäudeplanung, war ein Verlag angegliedert, der vorwiegend Literatur zu den Themen der Kommunikation und Organisationskybernetik publizierte. Vgl. Tügel 1968, 9-13.



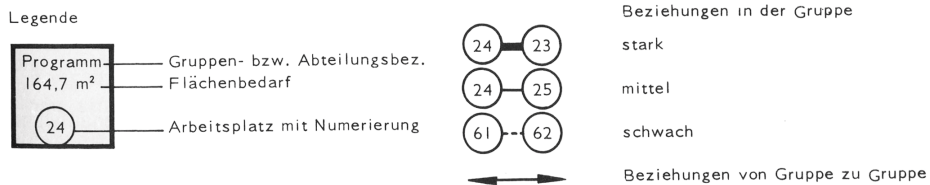
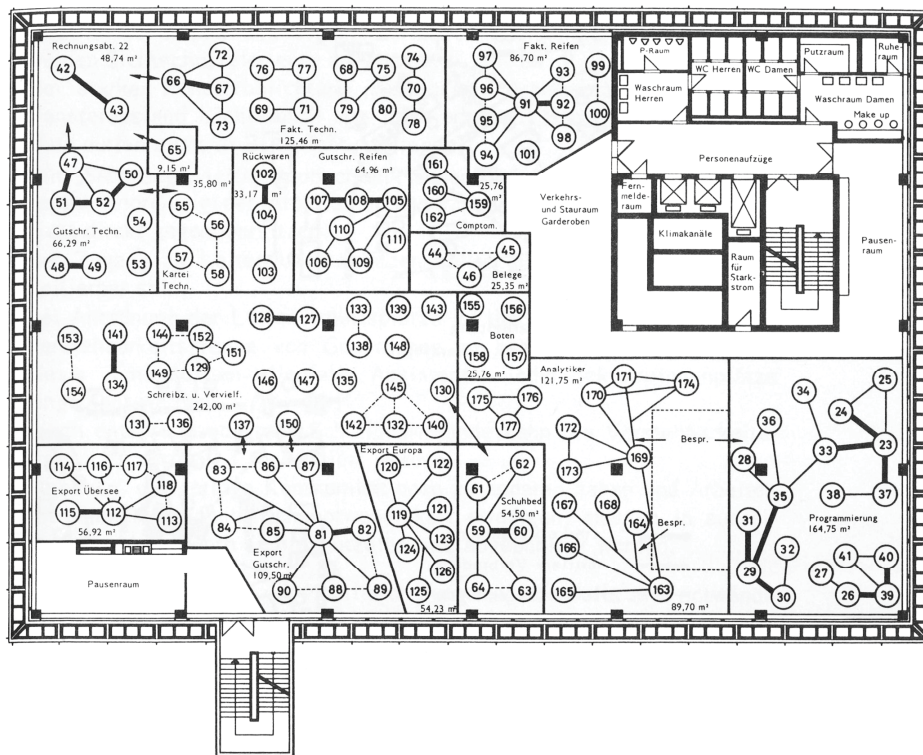
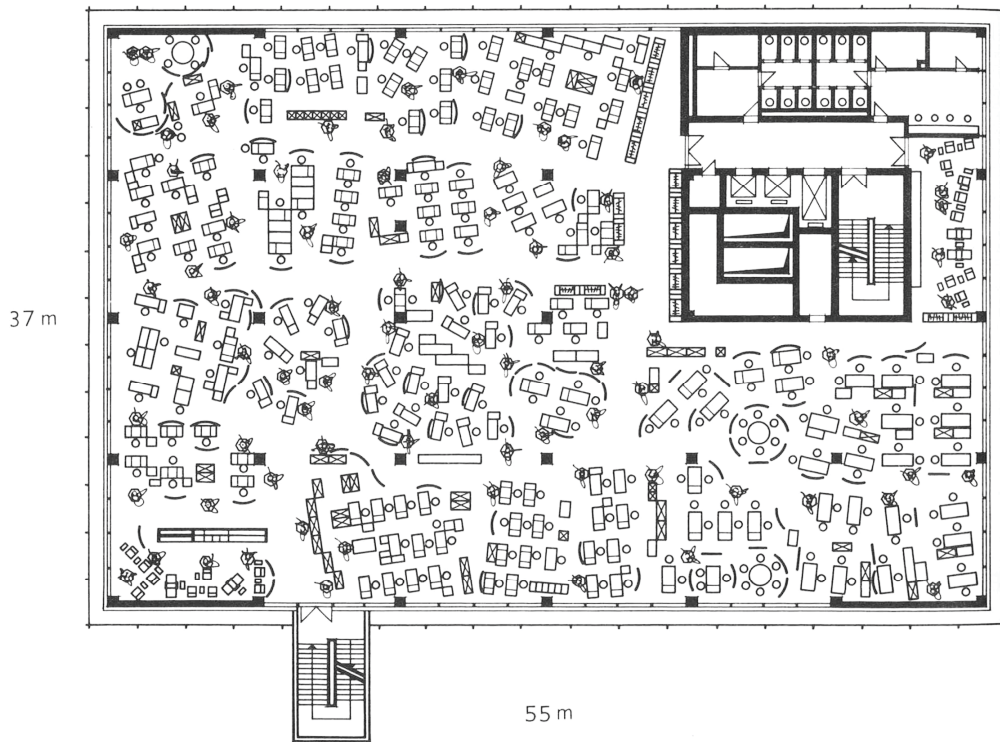


Abb 2.15 Gottschalk 1968, Beziehungen zwischen Arbeitsplätzen

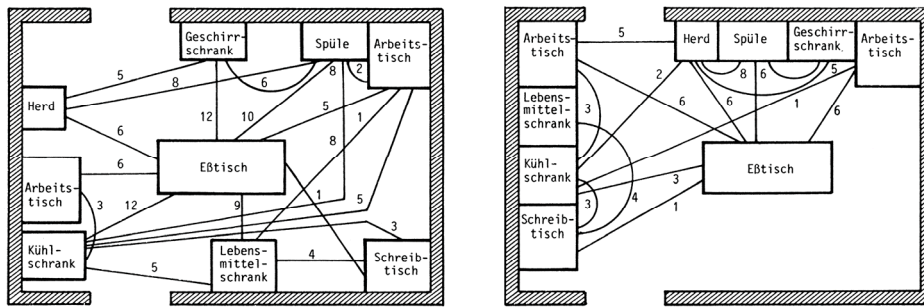


Abb 2.16 Kantowitz &amp; Sorkin 1983, Verknüpfungsanalyse Küche

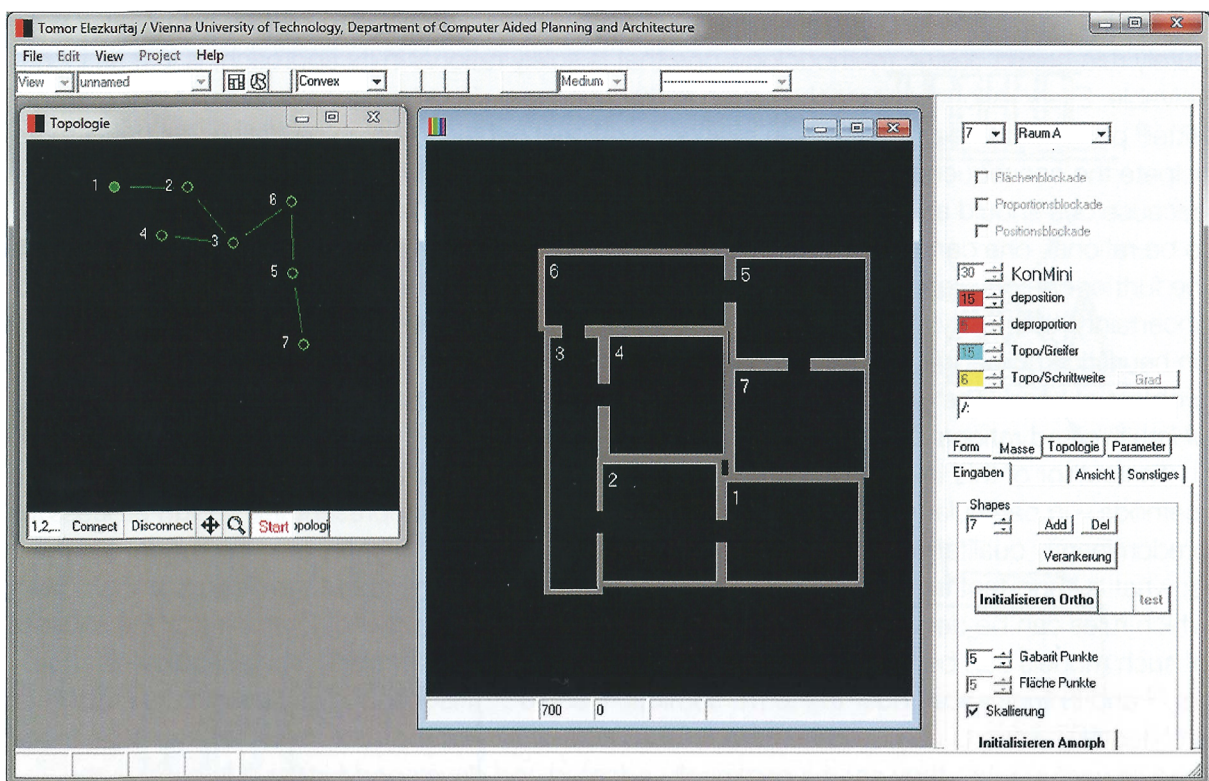


Abb 2.17 Tomor Elez Kurtaj 2002, Screenshot Floor Plan Program

Abbildung 2.16 zeigt die Struktur des Systems einer Küche vor und nach der Neuordnung der Komponenten auf Grundlage einer Verknüpfungsanalyse. Die Ziffern geben die Häufigkeit der Wege zwischen den Komponenten des Systems wieder. Auf ähnliche Weise wurden von Stanley Lippert im Jahre 1971 die Wege von Schwestern in Krankenstationen untersucht.<sup>25</sup>

Ein weiteres Forschungsfeld beim Einsatz von Graphen im Bereich der Architektur stellen die zahlreichen, seit den 1960er Jahren unternommenen Versuche dar, Gebäudeentwürfe

<sup>25</sup> Vgl. Kannheiser 1990, 74f.

maschinell zu generieren. Häufig dienten hierbei Graphen zur Definition der topologischen Randbedingungen. Beispiele ähnlicher Konzepte sind etwa Ulrich Flemings Computerprogramm DIM<sup>26</sup>, das Experiment YONA<sup>27</sup> von Nicholas Negroponte und Guy Weinzapfel oder jüngere Arbeiten von Geork Frank und Tomor Elezkurtay<sup>28</sup> (Abb 2.17) sowie der Züricher Forschungsgruppe *Kaisersrot*<sup>29</sup>.

## 2.5 Schlussfolgerungen

Die Graphentheorie ist heute besonders in der Informatik von großer Bedeutung. Mit Hilfe von Graphen lassen sich zahlreiche im Alltag auftretende Probleme modellieren und können auf diese Weise einer Lösung mit Hilfe von Algorithmen zugänglich gemacht werden. Bekannte Beispiele stellen Geo Informations Systemen (GIS) oder Navigationssoftware dar. Die Untersuchung komplexer Netzwerke mit Hilfe von Graphen stellt ein äußerst aktuelles und lebhaftes Forschungsgebiet dar, welches in zahlreichen wissenschaftlichen Disziplinen Anwendung findet. Neben der Analyse von realen Computernetzwerken werden die Konzepte der Netzwerktheorie beispielsweise auch in der Physik, der Biologie, der Klimaforschung oder der Soziologie angewandt. Wie schon anhand der Organigramme in Kapitel 1 gezeigt wurde, stellen Graphen auch ein geeignetes Mittel für eine diagrammatische Formulierung der relationalen Inhalte von Raumprogrammen dar. Im Informationsdesign hat sich der Einsatz von Graphen zur Visualisierung komplex vernetzter Systeme zu einer eigenen Kunstform entwickelt.

Topologie war von Anfang an mit dem städtischen Raum verbunden und in der Geburtsstunde der Graphentheorie war sogar eine konkrete Stadt Anlass und Gegenstand der Analyse.<sup>30</sup> Eulers Lösung des Königsberger Brückenproblems aus dem frühen 18. Jahrhundert weist bereits grundlegende Wesenszüge moderner Datenmodellierung und Programmierung auf.<sup>31</sup> Das Merkmal der Stadt einen Rundweg anzubieten, der über jede der sieben Brücken genau einmal führt, wurde durch das von Euler entwickelte Verfahren in allgemeiner Form operationalisiert. Während die Königsbergerinnen und Königsberger auf empirischen Wegen nach einem Rundgang mit besonderen Eigenschaften suchten, galt das Interesse des Mathematikers vielmehr der Entwicklung einer allgemeinen Routine zur Bestimmung des gefragten Merkmals a priori und in einer beliebigen Stadt.

*»Euler [...] fragte grundsätzlich oder eben, wie es ein Königsberger Philosoph wenig später nennen wird: transzendental. Emblematisch für die Topologie ist nicht nur der Umstand dass Euler nahezu blind war, sondern vor allem auch, dass er gar nicht vor Ort war, als er das Königsberger Brückenproblem von*

---

<sup>26</sup> Vgl. Flemming 1975.

<sup>27</sup> Vgl. Negroponte/Weinzapfel 1976, 74-78.

<sup>28</sup> Vgl. Valena u.a. 2011, 277ff.

<sup>29</sup> [www.kaisersrot.ch](http://www.kaisersrot.ch)

<sup>30</sup> Vgl. Günzel 2008, 9.

<sup>31</sup> Als weltweit erstes Computerprogramm gilt Ada Lovelace's etwa hundert Jahre später (1843) niedergeschriebene Handlungsanweisungen zur Berechnung von Bernoulli-Zahlen. Auch ihre Vorschriften konnte nur von Hand ausgeführt werden, da die bereits im 19. Jahrhundert entwickelte *Analytical Engine* von Charles Babbage, für die das Programm geschrieben war, zu diesem Zeitpunkt noch nicht gebaut werden konnte. Vgl. Wikipedia, dt.: Ada Lovelace, 01.11.2014.

*seinem Schreibtisch in Petersburg aus löste: Denn die konkrete architektonische Situation war transformierbar in eine relationale Raumbeschreibung, die sich von der materiellen Topographie löste, welche den Blick auf das Entscheidende geradezu verdeckte. Die Topologie negiert damit nicht die Materialität oder die topographischen Gegebenheiten, sondern hebt sie auf in eine Beschreibung wesentlicher Elemente: Bei Euler waren dies die Momente der ‚Knoten‘ und ‚Kanten‘, also der Verbindungslinien und Kreuzungspunkte. An diesen ist nicht entscheidend, wo sie geographisch lokalisiert sind, sondern welche Möglichkeiten sie dem Nutzer der Stadt in räumlicher Hinsicht bieten.«<sup>32</sup>*

Wie hier bereits vom Medientheoretiker Stefan Günzel angedeutet, weist die von Euler angewandte Methode zur Lösung des Königsberger Brückenproblems Parallelen zu dem Begriff des *Transzendentalen* nach Immanuel Kant auf. Der in Königsberg geborene Philosoph war zu diesem Zeitpunkt gerade sechs Jahre alt. Ob Kant Eulers Lösung des Königsberger Brückenproblems gelesen hat, ist nicht belegt. Das Studium Eulers Abhandlungen über Raum und Zeit sollte das Denken des Philosophen Jahre später in jedem Fall noch entscheidend beeinflussen. Über unterschiedliche Facetten des Zusammentreffens und Zusammengehens des Philosophen Kant mit dem Mathematiker Euler berichtet Heinrich Emil Timerding ausführlich in seinem Aufsatz *Kant und Euler* aus dem Jahre 1919.<sup>33</sup> Zwei Jahre vor Eulers Tod, im Jahre 1783, veröffentlichte der damals noch relativ unbekannte Philosoph seine *Kritik der reinen Vernunft*, in der er den Grundriss der Transzendentalphilosophie liefert.

*»Ich nenne alle Erkenntnis transzendental, die sich nicht so wohl mit Gegenständen, sondern mit unserer Erkenntnisart von Gegenständen, so fern diese a priori möglich sein soll, überhaupt beschäftigt. Ein System solcher Begriffe würde Transzendental-Philosophie heißen.«<sup>34</sup>*

Euler leitete aus der Untersuchung des speziellen Falls in Königsberg (induktiv) einen allgemeinen Erkenntnisprozess ab. Das strikte Befolgen seiner Anweisungen in sechs Teilschritten führt immer zu einem Ergebnis. Dabei ist es gleichgültig um welche Stadt es sich handelt und wie sich deren topografische Situation darstellt. Die Handlungsvorschrift zur allgemeinen Lösung des *Brückenproblems* (siehe Seite 34) für eine beliebige Stadt trägt die Grundzüge eines Algorithmus. Eulers Vor-Schrift war ursprünglich an Menschen gerichtet. Der Mathematiker programmierte die menschlichen Handlungen, welche zur Identifikation des gesuchten Merkmals notwendig sind. Auch ein Computer ist heute in der Lage, die einzelnen klar definierten Schritte durchzuführen, und schlussendlich zu einem Urteil über das Vorhandensein einer *Eulertour* zu kommen. Die Zergliederung der Gesamtoperation in einzelne explizite Handlungsschritte entspricht im Wesentlichen der Definition von Programmierung nach Alan Mathison Turing aus dem Jahre 1950:

---

<sup>32</sup> Günzel 2008, 9.

<sup>33</sup> Vgl. Timerding 1919.

<sup>34</sup> Kant [erstveröffentlicht 1781] 2011, 72.

*»Will man eine Maschine dazu bringen, bei einer schwierigen Operation das Verhalten des menschlichen Rechners nachzuahmen, muss man ihn fragen, wie sie ausgeführt wird, und die Antwort dann in Form einer Befehlsliste übertragen. Die Aufstellung von Befehlslisten bezeichnet man gewöhnlich als „programmieren“.«<sup>35</sup>*

Der Medienwissenschaftler Bernhard Dotzler vergleicht diesen Vorschlag Turings in seinem Aufsatz *Zur Archäologie der Computerkultur* mit einer informationstechnischen oder kybernetischen Übersetzung der von Kant vorgenommenen „Zergliederung des Verstandesvermögens“<sup>36</sup>, durch die Implementierung eines Computerprogramms, das den von Kant analysierten Erkenntnisfunktionen isomorph ist. Da Daten immer nur empirischer Natur sein können, schließt Dotzler, dass auch das Ergebnis jeder Datenverarbeitung zweifellos nur a posteriori sein kann. Anders verhält es sich jedoch mit der Verarbeitung der Daten als solche: Diese geschieht laut Dotzler nach Gesetzen a priori, nach Regelschritten, die bereits vor jeder Aufnahme von Daten festgelegt sein müssen.

*»Das Erkenntnisermöglichende der Turing-Maschine ist ihr Programm. In ihm liegen die Transzendentalien, gemäß denen sie notwendig denkt.«<sup>37</sup>*

Reale, räumliche Phänomene lassen sich durch die Schaffung geeigneter Modelle auf mathematischem Weg nachweisen. In Modellwelten können Erkenntnisse mit Hilfe von Algorithmen gewonnen werden. Euler bildete ein topologisches Modell der Stadt Königsberg. Die Abstraktion erfolgte durch Weglassen aller unnötigen Details. Da sich das verkürzte Abbild in den untersuchten Belangen entsprechend der Wirklichkeit verhielt, war es Euler durch eine Analyse des Modells möglich, Aussagen über die Realität zu treffen und sogar gestalterisch in diese einzugreifen. Der Mathematiker und Kulturwissenschaftler Wladimir Velminski hebt hervor, dass Eulers Vorschlag eine achte Brücke an der von ihm markierten Stelle (Abb 2.4a) zu errichten, um das Problem des Rundgangs zu lösen, das erste Beispiel einer Anordnung des Raums darstellt, deren „spezifische Facetten parallel zu der topologischen Wissensinformation verlaufen“<sup>38</sup>.

---

<sup>35</sup> Turing 1950, 155.

<sup>36</sup> Kant [erstveröffentlicht 1781] 2011, Analytik der Begriffe, 118.:

*Ich verstehe unter der Analytik der Begriffe nicht der Analysis derselben oder das gewöhnliche Verfahren in philosophischen Untersuchungen, Begriffe, die sich darbieten, ihrem Inhalte nach zu zergliedern und zur Deutlichkeit zu bringen, sondern die noch wenig versuchte Zergliederung des Verstandesvermögens selbst, um die Möglichkeit apriori dadurch zu erforschen, daß wir sie im Verstande allein, als ihrem Geburtsort, aufsuchen und dessen reinen Gebrauch überhaupt analysieren: denn dieses ist das eigenthümliche Geschäft einer Transzendental-Philosophie; das übrige ist die logische Behandlung der Begriffe in der Philosophie überhaupt.*

<sup>37</sup> Dotzler 2006, 45.

<sup>38</sup> Velminsky 2009, 168.

## 3 FORMALISIERTE GEBÄUDESPEZIFIKATION

Die Raumprogramme, welche heute in der Praxis des Entwerfens Verwendung finden, stellen eine komplexe Sammlung vernetzter Informationen über die Anforderungen an ein zukünftiges Gebäude dar. Zur Erstellung solcher Programme werden bislang vor allem Textverarbeitungs- oder Tabellenkalkulationsprogramme eingesetzt. Die Speicherung erfolgt dabei in Dokumentenformaten, welche ausschließlich von Menschen gelesen und verstanden werden können. Im folgenden Kapitel wird das Konzept einer formalisierten Gebäudespezifikation vorgestellt, das heißt ein in strenger Schriftform verfasstes Raumprogramm, dessen Zeichen in einer festgelegten Reihenfolge auf festgelegte Art und Weise verarbeitet werden. Das Ziel der Formalisierung besteht darin, die in heutigen Raumprogrammen enthaltenen Informationen durch eine geeignete Modellierung auch maschinell verwertbar zu machen.

Nach einer Einführung in die Datenmodellierung, erfolgt zuerst eine allgemeine Betrachtung des Merkmalsbegriffs in Informatik, Philosophie und Statistik. Im Anschluss daran wird einerseits beschrieben, wodurch sich qualitative von quantitativen Merkmalen unterscheiden, und es werden die in den empirischen Wissenschaften üblichen Skalenniveaus vorgestellt. Danach wird gezeigt, wie Zielvorstellungen bezüglich der Ausprägungen interessierender Gebäudemerkmale mit Hilfe sogenannter Präferenzfunktionen abgebildet werden können. Aufbauend auf diesen Grundlagen wird die EXTENSIBLE SPACE ALLOCATION LANGUAGE (XSAL), der Vorschlag eines Standards zur Modellierung von strukturierten Gebäudespezifikationen mit Hilfe der Metasprache XML beschrieben. Am Ende des Kapitels wird der DESIGN GOAL MANAGER vorgestellt. Dieser vom Autor entwickelte grafische Editor ermöglicht eine komfortable kartografische Erstellung differenzierter Gebäude-Setups.

### 3.1 Datenmodellierung

Sollen Informationen über einen bestimmten Anwendungsbereich von Computern erfasst und weiterverarbeitet werden, kann dies durch die Erstellung von Datenbanken erfolgen. In Datenbanken können Teilbereiche der Realität abstrakt abgebildet werden. Diese werden in der Fachsprache als *Weltausschnitt* (slice of reality) bezeichnet. Reale Objekte oder Phänomene werden darin durch Informationsträger, den sogenannten *Entitäten* erfasst. Beim Erstellen einer Datenbank wird der interessierende Weltausschnitt analysiert und ein entsprechendes Datenmodell erstellt, dessen Aufbau durch ein exakt definiertes Schema festgelegt wird. Dieser Vorgang wird als *Datenmodellierung* bezeichnet. Da von der vielschichtigen Realität dabei immer nur jene Strukturen aufgenommen werden, die für die jeweilige Anwendung benötigt werden, ist damit unwillkürlich immer eine mehr oder weniger starke Abstraktion verbunden.

Datenmodelle können strukturell unterschiedlich aufgebaut sein. In der Datenbanktheorie unterscheidet man beispielsweise zwischen hierarchischer, relationaler oder objektorientierter Modellierung. In zweierlei Hinsicht sind diese unterschiedlichen Konzepte jedoch gleich: Erstens gehen alle Datenmodelle im Grunde von Objekten und Beziehungen aus, die im betreffenden Weltausschnitt gesucht und beschrieben werden. Zweitens sind alle gängigen Datenmodelle attributbasiert, das heißt sie erfassen den Anwendungsbereich (Weltausschnitt) durch die Zuweisung von Merkmalen (Attributen) zu Objekten. Den unterschiedlichen Objekttypen (Entitäten) werden bestimmte Attribute als Elemente zugeordnet, welche Informationsdetails über ein Objekt enthalten. Auf diese Weise wird die Struktur eines Objektes (z.B. eines Raumes) modelliert. Jedes Attribut trägt eine Bedeutung (z.B. „Größe“ oder „Orientierung“), die Daten über ein Objekt werden als konkrete Ausprägungen gespeichert (z.B. 100m<sup>2</sup> oder „gut“). Die Objekte einer Datenbank werden durch die Gesamtheit ihrer Attributwerte repräsentiert.<sup>1</sup>

## Merkmale

Der Ausdruck *Merkmal* bezeichnet eine Eigenschaft (lat. attributum) eines Gegenstandes<sup>2</sup> und hat die semantische Bedeutung eines Charakteristikums, mit dem man ein Einzelding versieht, um es unter mehreren Dingen, mit ansonsten eventuell gleichen Eigenschaften, auszuzeichnen. Das kann beispielsweise zum Zweck des Wiedererkennens oder Vergleichens erfolgen. In der Philosophie wird der Ausdruck *Merkmal* am häufigsten im Sinne eines Bestandteils von einem Begriff verwendet.<sup>3</sup>

*»Ein Merkmal ist dasjenige an einem Dinge, was einen Teil der Erkenntnis desselben ausmacht; oder - welches dasselbe ist - eine Partialvorstellung, sofern sie als Erkenntnisgrund der ganzen Vorstellung betrachtet wird. - Alle unsre Begriffe sind demnach Merkmale und alles Denken ist nichts anders als ein Vorstellen durch Merkmale.«<sup>4</sup>*

In den empirischen Wissenschaften spricht man von Merkmalen im Sinne einer erhobenen Größe beziehungsweise einer statistischen Variable. Im Rahmen empirischer Untersuchungen werden Daten erhoben, die mit Hilfe statistischer Methoden dargestellt und analysiert werden sollen. Auf diese Weise können inhaltliche Fragestellungen mathematisch erfasst und ausgewertet werden. Die Objekte, an denen die Daten beobachtet werden, nennt man *statistische Einheiten*. Die Menge aller *statistischen Einheiten*, über die man Aussagen gewinnen will, wird als *Population* bezeichnet. An den statistischen Einheiten werden interessierende Größen beobachtet, welche als *Merkmale* oder *Variablen* bezeichnet werden. Statistische Einheiten werden daher auch als *Merkmalsträger* bezeichnet. In der Regel wird an einem Objekt mehr als nur ein Merkmal erhoben. Jedes Merkmal kann verschiedene Werte annehmen, die sogenannten *Merkmalsausprägungen*. Der Begriff mit dem ein bestimmtes

---

<sup>1</sup> Vgl. Staud 2005, 18f.

<sup>2</sup> Hierbei kann es sich auch um eine Sache, eine Person oder einen abstrakten Zusammenhang handeln.

<sup>3</sup> Vgl. Gabriel, Merkmal, 1153.

<sup>4</sup> Kant 1800, 77f.

Merkmal bezeichnet wird (z.B. Orientierung, Größe) bildet eine Variable und beschreibt die interessierende Eigenschaft allgemein. Die Merkmalsausprägung, welche bei der Datenerhebung festgestellt wird (gut, schlecht, 30m<sup>2</sup>, 75m<sup>2</sup>), gibt den konkreten Wert dieser Variablen für eine bestimmte statistische Einheit an.<sup>5</sup> Eine Variable ist im Grunde nichts anderes als ein Symbol, das durch jedes Element einer spezifizierten Menge von Merkmalsausprägungen ersetzt werden kann.<sup>6</sup> Merkmale können anhand bestimmter Besonderheiten in verschiedene Typen eingeteilt werden. Eine Kategorisierung kann aufgrund der Anzahl und der Eigenschaften ihrer möglichen Ausprägungen erfolgen.

<i>Objekt</i>	<i>Orientierung</i>	<i>Größe</i>
<i>Raum</i>	<i>Gut</i>	<i>75 m<sup>2</sup></i>

### 3.2 Merkmalstypen

Merkmale lassen sich grundsätzlich in qualitative und quantitative Typen unterteilen. Für den wissenschaftlichen Gebrauch erfolgt innerhalb dieser Grundkategorien häufig eine differenziertere Einteilung in *Skalenniveaus*. Diese Einteilung der Merkmalstypen basiert im Wesentlichen auf den Eigenschaften der Skala, die zur Messung oder Erhebung des betreffenden Merkmals benutzt wird. Eine Skala gibt die möglichen Merkmalsausprägungen vor und repräsentiert eine Vorschrift welche die Zuordnung dieser Werte zu einem beobachteten Gegenstand vorgibt. Der Beobachtungswert gibt die Ausprägung des jeweils interessierenden Merkmals an. Obwohl Skalen im ursprünglichen Sinn nur numerische Werte liefern, ist es durchaus üblich für qualitative Merkmale auch Skalen zu benutzen, deren Werte Begriffe sind.<sup>7</sup> In der Literatur finden sich umfassende Klassifizierungen von Skalenniveaus.<sup>8</sup> Aus diesen sollen für die nachfolgenden Untersuchungen fünf elementare Typen herausgehoben werden: Nominal- und Ordinalskala für qualitative Merkmale, sowie Intervall-, Verhältnis- und Absolutskala für kardinale, quantitative Merkmale.

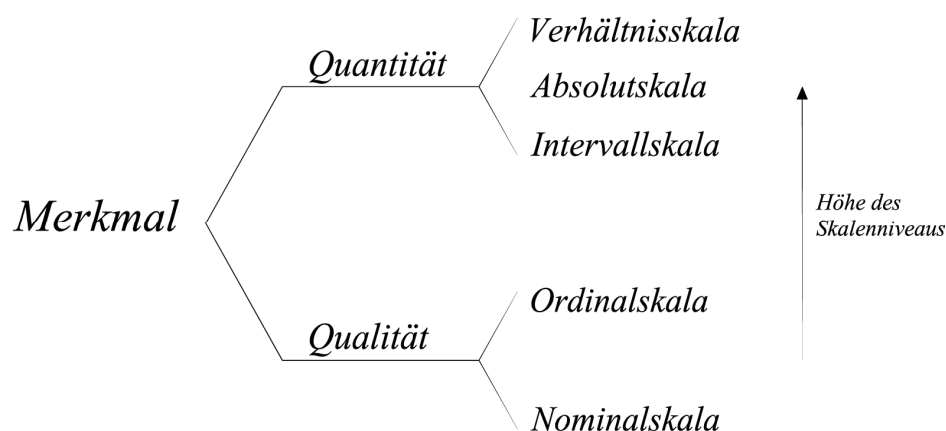


Abb 3.1 Merkmalstypen

<sup>5</sup> Vgl. Fahrmeir u.a. 2011, 16f.

<sup>6</sup> Vgl. Bortz 2006, 2.

<sup>7</sup> Vgl. Cramer/Kamps 2008.

<sup>8</sup> Vgl. Stevens 1946, Narens 1968, Orth 1974, Marks 1974.



## Qualitative Merkmale

Qualitative Merkmale besitzen endlich viele Ausprägungen in Form von Namen oder Kategorien und beschreiben (ihrer Bezeichnung entsprechend) Qualitäten. Qualitative Merkmale mit nur zwei möglichen Ausprägungen, wie zum Beispiel verbunden-separiert oder hell-dunkel, werden als *dichotom* bezeichnet. Merkmale mit mehreren möglichen Ausprägungen, wie beispielsweise das Merkmal der *Grundrissfigur eines Raums*, bei dem zwischen Kategorien wie dreieckig, rechteckig, kreisförmig unterschieden werden kann, nennt man *polytom*. Qualitative Merkmale haben keine Einheiten. Bei der Erhebung wird festgestellt, zu welcher Kategorie der untersuchte Gegenstand (Merkmalsträger, Untersuchungseinheit) gehört. Laut Jürgen Bortz müssen bei der Wahl dieser Kategorien drei wichtige Kriterien erfüllt werden.

1. Das *Genauigkeitskriterium*: Jede Kategorie muss exakt definiert sein. Dazu ist eine präzise Definition der operationalen Indikatoren für die einzelnen Kategorien des Merkmals erforderlich (vgl. Kapitel 5.1 Explikation). Nur wenn diese Indikatoren festgestellt werden, so fällt das untersuchte Objekt in die entsprechende Merkmalskategorie. Indikatoren für die Kategorie *rechteckig* zum Beispiel sind die Existenz von vier Ecken mit einem Winkel von jeweils 90°.
2. Das *Exhaustivitätskriterium*: Jedes untersuchte Objekt muss einer Merkmalskategorie zugeordnet werden können. In manchen Fällen ist es notwendig, den eigentlichen Kategorien eines Merkmals eine Hilfskategorie wie *Sonstige(s)* beizufügen. Wenn zum Beispiel beim Merkmal der *Figur* eines Polygons nur zwischen den Kategorien *dreieckig* und *rechteckig* unterschieden wird, ist für alle anderen Figuren, wie beispielsweise für ein Fünfeck, eine dritte Kategorie notwendig.
3. Das *Exklusivitätskriterium*: Die Kategorien müssen sich gegenseitig ausschließen, das heißt ein Objekt darf nicht gleichzeitig mehreren Kategorien eines Merkmals zugeordnet werden können. Die *Figur* eines Polygons kann beispielsweise nicht gleichzeitig ein Rechteck und ein Dreieck sein, jedoch gleichzeitig ein Viereck und ein Quadrat.<sup>9</sup>

## Skalenniveaus qualitativer Merkmale

Qualitative Merkmale können auf zwei wesentlichen Skalenniveaus erhoben werden. Die *Nominalskala* stellt das niedrigste Skalenniveau quantitativer Merkmale dar, und ist im Grunde eine Aneinanderreihung dichotomer Merkmalsausprägungen. Auf dieser Skala besteht zwischen den einzelnen Merkmalsausprägungen keine Rangordnung. Eine Reihung unterschiedlicher Ausprägungen nach der Größe ist nicht möglich.<sup>10</sup> Beim Vergleich zweier Merkmalsausprägungen können auf diesem Skalenniveau nur Aussagen über Gleichheit

---

<sup>9</sup> Vgl. Bortz 2006, 140.

<sup>10</sup> Vgl. Fahrmeir 2011, 17.

oder Ungleichheit gemacht werden. Die Ausprägungen nominalskalierter Merkmale werden durch Feststellung erhoben.<sup>11</sup>

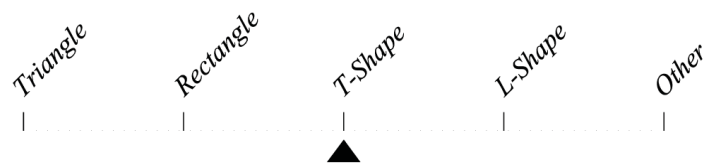


Abb 3.2 Nominalskala

Im Gegensatz dazu werden Merkmalsausprägungen auf *Ordinalskalenniveau* durch Vergleiche erhoben. Zusätzlich zu den Aussagen der Nominalskala besteht zwischen zwei Merkmalsausprägungen auf dieser Skala eine Rangordnung, das heißt unterschiedliche Ausprägungen können der Größe nach geordnet werden. Merkmale auf diesem Skalenniveau beschreiben zwar prinzipiell qualitative Eigenschaften, haben jedoch bereits einen zumindest schwachen quantitativen Aspekt. Eine Interpretation der Abstände zwischen zwei Ausprägungen ist jedoch nicht sinnvoll möglich.<sup>12</sup>

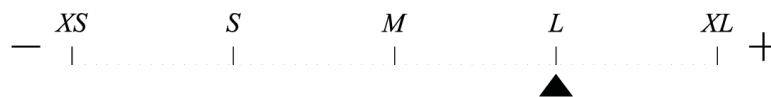


Abb 3.3 Ordinalskala

## Quantitative Merkmale

Quantitative Merkmale spiegeln Intensitäten oder Ausmaße in Form von Zahlenwerten wider und werden durch Messung im herkömmlichen Sinn, das heißt, durch den Vergleich mit einer Einheit erhoben. Die Ausprägungen von quantitativen Merkmalen können miteinander verglichen und der Größe nach gereiht werden. Im Unterschied zu qualitativen Merkmalen kann der Abstand zwischen zwei Ausprägungen (der ebenfalls eine Zahl ist) interpretiert werden. Man spricht in diesem Zusammenhang vom Vorhandensein einer Äquidistanz, das heißt, dass ein bestimmter Zahlenabstand immer den gleichen Qualitätsunterschied in der Merkmalsausprägung abbildet.<sup>13</sup> Dadurch sind Aussagen wie „A ist um X größer oder besser als B“ möglich.<sup>14</sup>

Anhand der Anzahl der möglichen Ausprägungen werden quantitative Merkmale in stetige und diskrete Typen eingeteilt. Als stetig bezeichnet man Merkmale deren Ausprägungen einen beliebigen, reellen Wert (in einem Zahlenintervall) annehmen können. Die Anzahl der möglichen Ausprägungen stetiger Merkmale ist somit in jedem Fall unendlich. Zwischen zwei benachbarten Ausprägungen kann immer noch eine weitere liegen (z.B. die Nutzfläche eines Raums). Ein Merkmal heißt hingegen diskret, wenn seine möglichen Ausprägungen abzählbar sind. Die Anzahl der möglichen Ausprägungen eines diskreten Merkmals kann endlich (begrenzt Intervall) oder abzählbar-unendlich sein. Im zweiten Fall hat die

<sup>11</sup> Vgl. Fahrmeir et al. 2011, 17f.

<sup>12</sup> Vgl. Ebd.

<sup>13</sup> Vgl. Rasch u.a. 2010, 11.

<sup>14</sup> Vgl. Fahrmeir u.a. 2011, 19.

Menge seiner möglichen Ausprägungen die gleiche Mächtigkeit, wie die der natürlichen Zahlen. Die Anzahl der Fenster stellt zum Beispiel ein diskretes Merkmal eines Raums dar.

## Skalenniveaus quantitativer Merkmale

Die *Intervallskala* stellt das niedrigste Skalenniveau quantitativer Merkmale dar. Auf dieser Skala gibt es keinen natürlichen Nullpunkt. Ein Beispiel für dieses Skalenniveau stellt die Temperatur gemessen in Grad Celsius dar. Der natürliche Nullpunkt ist jener Ort einer Skala, an dem die Variable aufhört zu existieren. Null Grad Celsius ist jedoch nicht mit *keiner* Temperatur gleichzusetzen. Auch der Intelligenzquotient, die meisten Ratingskalen, Jahreszahlen und Zeitpunkte zählen zu intervallskalierten Merkmalen.<sup>15</sup>

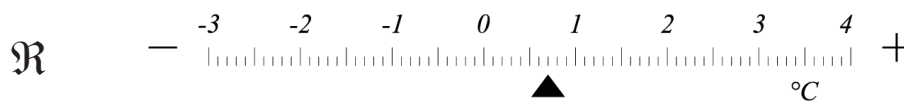


Abb 3.4 Intervallskala

Die *Verhältnisskala* ist das höchste Skalenniveau, das ein quantitatives Merkmal erreichen kann. Voraussetzung für dieses Skalenniveau ist die Existenz eines natürlichen Nullpunkts. Die Maßeinheit verhältnisskalierter Merkmale kann willkürlich definiert werden. Zusätzlich zu den Aussagen der Intervallskala sind auf diesem Skalenniveau auch Aussagen über Verhältnisse einzelner Merkmalsausprägungen möglich wie „A ist halb so groß wie B“.<sup>16</sup>



Abb 3.5 Verhältnisskala

Die *Absolutskala* hat im Grunde dieselben Eigenschaften wie die Verhältnisskala. Der Unterschied liegt nur darin, dass die Maßeinheit auf diesem Skalenniveau natürlich gegeben ist, das heißt, es geht im Wesentlichen um Stückzahlen, welche durch Zählung erhoben werden.<sup>17</sup>



Abb 3.6 Absolutskala

<sup>15</sup> Vgl. Fahrmeir u.a. 2011, 17f.

<sup>16</sup> Ebd.

<sup>17</sup> Cramer/Kamps 2008, 10.

## Zusammenfassung

Die Einteilung in Skalenniveaus stellt eine hierarchische Ordnung dar, vom niedrigsten Niveau der Nominalskala bis hinauf zur Verhältnisskala. Merkmale, die auf einem hohen Skalenniveau gemessen wurden, können nachträglich auf ein niedrigeres Niveau transformiert werden. Umgekehrt ist eine Transformation von einem niedrigeren auf ein höheres Niveau nicht möglich. Bei einer Senkung des Skalenniveaus von erhobenen Daten geht immer Information verloren.<sup>18</sup>

Betrachten wir dazu folgendes Beispiel: Eine Wohnanlage beinhaltet Wohnungen unterschiedlicher Größe, von 40 bis 100m<sup>2</sup>. Bei der Wohnungsgröße handelt es sich um ein verhältnisskaliertes Merkmal. Es ist nun möglich die Wohnungen in nur drei Kategorien einzuteilen - in Wohnungen unter 60m<sup>2</sup>, Wohnungen zwischen 60 und 80m<sup>2</sup>, und Wohnungen größer als 80m<sup>2</sup>. Die drei Gruppen könnten als *groß*, *mittel*, und *klein* bezeichnet werden. Bei der Zuordnung jeder Wohnung zu einer Kategorie geht die genaue Quadratmeterzahl verloren. Die Wohnungsgröße ist jetzt nur noch ein ordinalskaliertes Merkmal, welches umgekehrt nicht mehr in ein verhältnisskaliertes Merkmal transformiert werden kann. Der Downgrading-Prozess ist in Abbildung 3.7 dargestellt.

Für jedes Merkmal ist die Entscheidung zu treffen, auf welchem Skalenniveau es erhoben werden soll oder kann. Allgemein betrachtet empfiehlt sich in empirischen Untersuchungen bei der Quantifizierung das höchstmögliche Skalenniveau zu wählen. Die Daten können später bei Bedarf auf ein niedrigeres Niveau transformiert werden.<sup>19</sup> Nicht immer ist mit dem höchsten Skalenniveau auch die höchste Aussagekraft verbunden. Einen Raum einfach als „sehr hell“ zu bezeichnen kann unter Umständen verständlicher sein als die Angabe eines präzisen Lux-Wertes zur Beschreibung der Helligkeit dieses Raums.

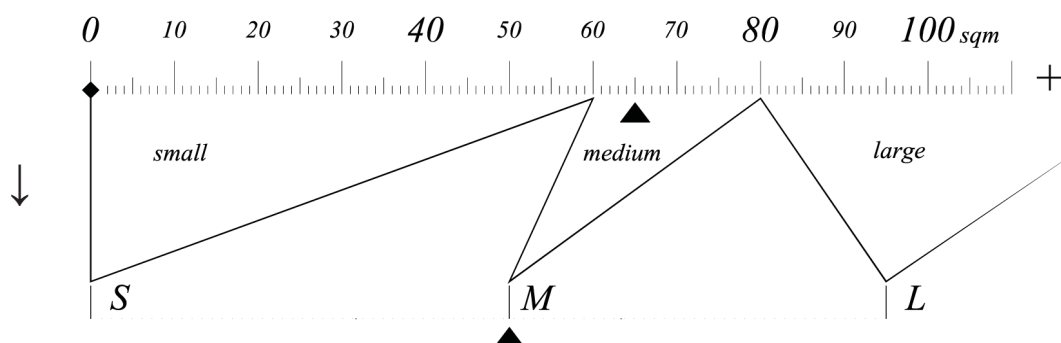


Abb 3.7 Transformation von der Verhältnisskala ( $\mathfrak{R}_+$ ) zur Ordinalskala (S,M,L)

<sup>18</sup> Vgl. Fahrmeir 2011, 18f.

<sup>19</sup> Vgl. Bortz 2006, 70.

### 3.3 Präferenzfunktionen

Nachdem das Skalenniveau eines interessierenden Gebäudemerkmals festgesetzt wurde, kann in einer Gebäudespezifikation definiert werden, welche Merkmalsausprägungen für die zukünftige Nutzung (z.B. eines Raums) notwendig oder vorteilhaft wären. Das Merkmal der Größe eines Raums könnte beispielsweise auf der Verhältnisskala mit der Einheit Quadratmeter erhoben werden. Für Raumgrößen erfolgt in heutigen Raumprogrammen üblicherweise die Zuweisung eines einzelnen Zielwertes. Von einem bestimmten Raum könnte eine Größe von  $25\text{m}^2$  gewünscht werden. Die Zielsetzung könnte jedoch auch in Form einer Ungleichung erfolgen. Die Angabe  $>25\text{m}^2$  würde bedeuten, dass der betreffende Raum mindestens  $25\text{m}^2$  groß sein soll.

Ähnlich verhält es sich mit Wünschen bezüglich qualitativer Merkmale. Die Grundrissfigur eines Raums kann beispielsweise auf der Nominalskala, mit den möglichen Ausprägungen rechteckig, dreieckig, kreisförmig, erhoben werden. Von einem bestimmten Raum kann (z.B. aufgrund der Möblierbarkeit) gefordert werden, dass dieser im Idealfall die Form eines Rechtecks aufweist, zumindest keine spitzen Winkel enthält, oder auf keinen Fall rund sein sollte. Damit die Evaluierung der Erfüllung solcher Teilziele maschinell erfolgen kann, müssen die Erwartungshaltungen durch ein geeignetes Modell mathematisch ausgedrückt werden. In der Entscheidungstheorie<sup>20</sup> erfolgt die Modellierung von Präferenzen durch die Ermittlung von sogenannten Präferenz- oder Wertfunktionen (Value Functions). Wertfunktionen bilden die Präferenzen eines Entscheiders aufgrund messtheoretischer Überlegungen ab. Sie werden in der Regel auf das Intervall  $[0,1]$  normiert.<sup>21</sup> Christoph Zangemeister, Professor am Institut für Psychologie und Arbeitswissenschaft an der TU Berlin, definiert eine solche Funktion mit folgenden Worten:

*»Eine Wertfunktion  $v$  ist eine Funktion, die jeder Alternative  $a$  eine reelle Zahl derart zuordnet, dass der Wert einer Alternative  $a$  genau dann größer als der Wert einer Alternative  $b$  ist, falls der Entscheider  $a$  gegenüber  $b$  präferiert.«<sup>22</sup>*

Das Resultat der Wertfunktion kann auch als Zielwert oder Erfüllungsfaktor bezeichnet werden. Der Wert 1.0 bedeutet immer, dass ein Ziel gänzlich erfüllt wurde. Der Wert 0.0 sagt aus, dass ein Ziel überhaupt nicht erfüllt wurde. Wenn es das Skalenniveau des interessierenden Merkmals zulässt, kann eine entsprechende Wertfunktion (Measurable Value Function) auch Erfüllungsfaktoren zwischen diesen Extremen liefern. Auf diese Weise kann die Stärke einer Präferenz graduell festgesetzt werden.

Eine mögliche Wertfunktion des verhältnisskalierten Merkmals der Größe eines bestimmten Raums könnte zum Beispiel auf folgende Weise festgesetzt werden: Der Erfüllungsfaktor soll genau dann 1.0 betragen, wenn die erhobene Ausprägung exakt dem Zielwert, beispielsweise  $35\text{m}^2$ , entspricht. Der Erfüllungsfaktor soll 0.0 betragen, wenn die erhobene Ausprägung um 5 Quadratmeter vom Ziel abweicht. Der Verlauf einer Wertfunktion, welche

---

<sup>20</sup> Die Entscheidungstheorie ist eine mathematische Disziplin, welche in der Betriebswirtschaft häufig als Instrument zur Ermittlung einer optimalen Entscheidung oder Lösung einer Problemstellung herangezogen wird.

<sup>21</sup> Vgl. Eisenführ 2010, 114.

<sup>22</sup> Eisenführ 2010, 111.

diesen Definitionen entspricht ist in Abbildung 3.8 dargestellt. Die Merkmalsausprägungen  $x$  werden horizontal auf der Abszisse aufgetragen, die Ordinate repräsentiert den jeweiligen Erfüllungsfaktor  $f(x)$ . Die Breite der Verteilungskurve drückt die Toleranz gegenüber dem Zielwert aus.

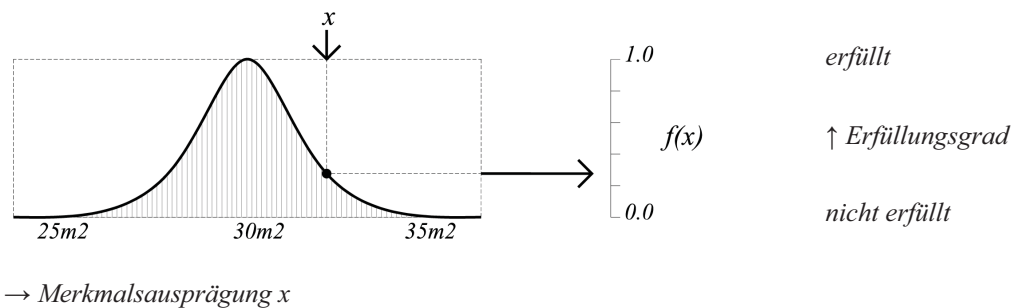


Abb 3.8 Beispiel Wertefunktion Raumgröße

Als zweites Beispiel betrachten wir das diskrete Merkmal *Belichtung*, welches gemäß der ARCHITEKTUR ROUTINE No.11 (s. Anhang) auf der Ordinalskala erhoben wird. Den möglichen Ausprägungen *good*, *medium* und *bad* werden numerische Werte zwischen 0 und 1 zugeordnet. Die Abstände zwischen den diskreten Ausprägungen werden gleichmäßig bewertet (vgl. Ratingskala Kapitel 3.2). Für *mittlere Belichtung* beträgt der Erfüllungsfaktor dieser Präferenzfunktion den Wert 0.5.

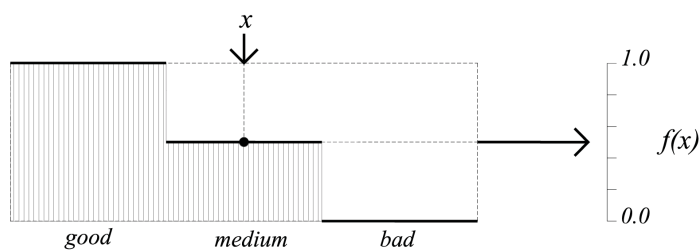


Abb 3.9 Beispiel Wertefunktion Belichtung

Bei einem nominalskalierten Merkmal kann ähnlich vorgegangen werden. Jeder Kategorie wird durch die Wertefunktion ein Zielerfüllungswert zugeordnet. Im vorliegenden Beispiel der Wertefunktion des Merkmals *Figur* liegt eine eindeutige Bevorzugung der Rechtecksform vor. Das Dreieck wird relativ schlecht bewertet, und der Kreis erhält den Wert 0.

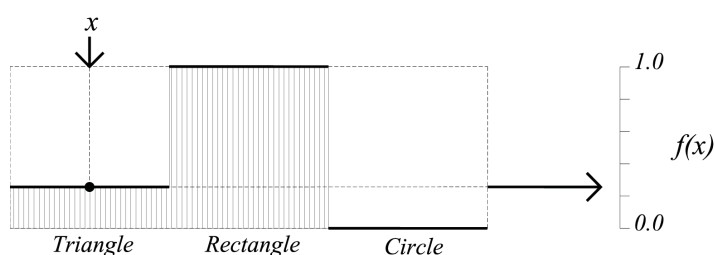


Abb 3.10 Beispiel Wertefunktion Figur

Die Abbildungen 3.11 und 3.12 zeigen beispielhaft weitere Wertfunktionen.

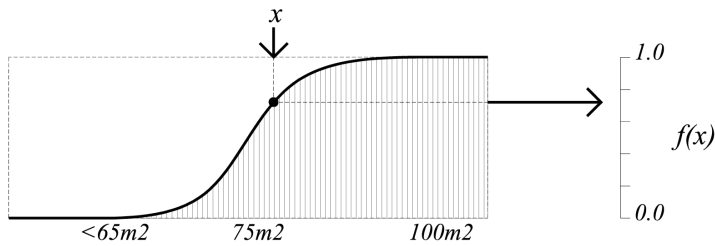


Abb 3.11 ‚Weiche‘ Ungleichung

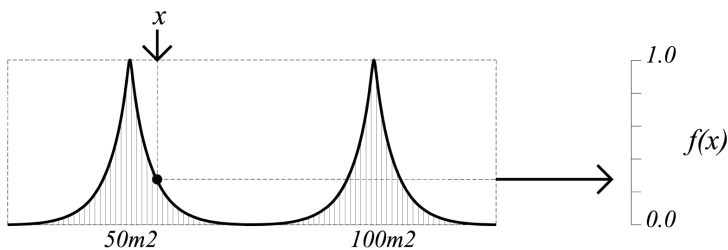


Abb 3.12 Mehrere Hochpunkte

## Bestimmung der Wertfunktion

Zur Bestimmung von Wertfunktionen werden in der Literatur unterschiedliche Methoden beschrieben.<sup>23</sup> Beispiele stellen die *Direct Rating Methode*, die *Methode gleicher Wertdifferenz* oder die *Halbierungsmethode* dar.<sup>24</sup> Zangemeister empfiehlt einerseits, wenn möglich zur Ermittlung von Präferenzfunktionen, auf mathematisch exakte Methoden zurückzugreifen, weist aber auch relativierend darauf hin, dass es sich bei der Modellierung von subjektiven Präferenzmustern im Grunde fast immer um freie Vereinbarungen handelt:

*»Da man bei normativen Entscheidungsmodellen nicht notwendig von einer empirisch nachvollziehbaren Funktionalbeziehung zwischen dem Objektkontinuum und dem subjektiven Präferenzkontinuum ausgehen muss, genügt es hier, eine Wertfunktion zugrunde zu legen, deren Charakteristik als rationale Entscheidungsgrundlage akzeptiert werden kann.«<sup>25</sup>*

Im Allgemeinen werden von Wertfunktionen zumindest zwei notwendige Eigenschaften gefordert: Erstens dass die mathematisch ausgedrückten Präferenzordnungen in jedem Fall *vollständig* sein müssen. Dies ist dann der Fall, wenn für jedes beliebige Alternativenpaar eine Präferenz bezüglich einer der beiden existiert. Eine zweite grundlegende Forderung besteht häufig darin, Präferenzordnungen *transitiv* zu gestalten. Transitivität besteht dann, wenn für drei beliebige Alternativen a,b,c, als Ergebnisse von  $f(x)$ , folgende Bedingung gilt: Aus  $a > b$  und  $b > c$  folgt  $a > c$ .<sup>26</sup>

<sup>23</sup> Vgl. Eisenführ 2010, 115.

<sup>24</sup> Vgl. Eisenführ 2010, 117ff.

<sup>25</sup> Zangemeister 1970, 219.

<sup>26</sup> Vgl. Eisenführ 2010, 109ff.

## Zusammenfassung

Mit Hilfe von Wertfunktionen ist die mathematische Modellierung differenzierter Zielvorstellungen möglich. In der Ausformulierung des Kurvenverlaufs besteht hohe Flexibilität und damit die Möglichkeit ein weites Spektrum an Präferenzmustern durch situationsrelevante Festlegungen auszudrücken (vgl. Kapitel 4.2 Paretoprinzip). Es ist sogar möglich eine Vorliebe für Achsen-, Raster- oder Modulormaße<sup>27</sup> durch eine Funktion mit entsprechenden, sich wiederholenden Hochpunkten zu modellieren (Abb 3.12). Ein Vorteil besteht auch darin, dass polygonale Wertfunktionen ohne Schwierigkeiten programmiert und ausgewertet werden können. Die Definition einer sinnvollen Präferenzfunktion erfordert durchaus etwas Geschick und kann gewissermaßen als Kunst bezeichnet werden.

## 3.4 Extensible Space Allocation Language - XSAL

Durch die generische Auszeichnungssprache XML (Extensible Markup-Language) kann nahezu jedes beliebige Datenmodell beschrieben werden. XML ist eine *Sprache zum Beschreiben von Sprachen* und somit eine Metasprache. Auf der Basis von XML können strukturell und inhaltlich eingeschränkte, anwendungsspezifische formale Sprachen definiert werden.<sup>28</sup> Jede XML-Sprache hat ihre eigene Grammatik, diese wird in Form eines XML-Schemas (z.B. XSD - XML Schema Definition) definiert. Ein Schreiben das in einer XML Sprache verfasst wurde, ist nur dann gültig, wenn dieses *wohlgeformt* (well-formed) ist, das Dokument den Verweis auf das zugrundeliegende Schema enthält und die darin enthaltenen Formatierungsvorschriften auch einhält. XML-Dokumente bestehen aus (gewöhnlichen) Textzeichen. Im Gegensatz zur Binärcodierung anderer Datenformate, bleibt der Inhalt eines XML Dokuments auf diese Weise einerseits für Menschen lesbar, zugleich können die Daten auch mit Hilfe eines XML-Parsers eingelesen werden, und stehen so auch für eine Weiterverarbeitung durch den Computer bereit. Die unterschiedlichsten Rechnersysteme können XML lesen und schreiben. Aus diesem Grund wird XML häufig auch als *Lingua Franca* bezeichnet, in Anlehnung an eine von den venezianischen Kaufleuten im Mittelmeerraum verwendete gemeinsame Verkehrssprache, einem mit arabischen Elementen vermischten Italienisch. Durch den erweiterten von Unicode- Zeichensatz können neben dem lateinischen Alphabet beispielsweise auch Japanische, Chinesische oder Arabische Schriftzeichen benutzt werden.<sup>29</sup> Neben der Plattformunabhängigkeit, macht gerade die Möglichkeit der gemischten Nutzung von menschlichem Leser und maschineller Verarbeitung die Auszeichnungssprache XML zu einem geeigneten Medium zur Modellierung von Gebäudespezifikationen. Hierzu wird im folgenden Unterkapitel der Vorschlag eines XML Standards mit der Bezeichnung EXTENSIBLE SPACE ALLOCATION LANGUAGE (XSAL) definiert und ein bereichsspezifisches Vokabular festgesetzt. In XSAL werden nur Gebäudeelemente beschrieben, welche auch Entitäten des in Kapitel 4 vorgestellten Gebäudedatenmodell

---

<sup>27</sup> Der *Modulor* ist ein vom Architekten Le Corbusier entwickeltes, auf dem Goldenen Schnitt basierendes Maßsystem.

<sup>28</sup> Ein Beispiel hierfür sind die XML-basierten Standards der Text Encoding Initiative (TEI). In den Geisteswissenschaften wird TEI zum Auszeichnen von natürlich-sprachlichen Texten benutzt. Mit Hilfe von TEI kann beispielsweise ein Gedicht oder Drama mit textkritischen Anmerkungen versehen oder die Bedeutung der verwendeten Ausdrücke klargestellt werden.

<sup>29</sup> Vgl. Vonhoegen 2011, 28ff.



darstellen. Durch die differenzierte Charakterisierung und Verknüpfung einer relativ kleinen Anzahl unterschiedlicher Objekttypen kann dabei ein komplexes System beschrieben werden.

## Gruppen

Begonnen und beendet wird ein XSAL Dokument durch das Wurzelement `<spaceallocation>`. Innerhalb des Wurzelements befinden sich alle weiteren Informationen. Die grundlegende Strukturierung des Raumprogramms durch Obergruppen erfolgt in XSAL durch das Element `<group>`. Jede Gruppe wird über das Attribut *name* mit einem Namen versehen. Jede Gruppe kann wiederum eine oder mehrere Untergruppen beinhalten.

```
<group name="Building A">
  <group name="Apartment 1">
    [...]
  </group>
  <group name="Apartment 2">
    [...]
  </group>
  [...]
</group>
```

Abb 3.13 Gruppen

## Räume & Raumzonen

Innerhalb eines `<group>` Elements kann die Definition der einzelnen Räume eines zukünftigen Gebäudes erfolgen. Ein Raum wird durch das Element `<room>` ausgedrückt. Innerhalb jedes `<room>` Elements können beliebig viele Nutzungsbereiche durch ein- oder mehrfache Deklaration des Elements `<zone>` definiert werden. Auch Regionen erhalten einen Namen als Werte des Attributs *name* des Elements `<zone>`.

```
<group name="Apartment 1">
  <room name="Living">
    <zone name="Watching TV">
      [...]
    </zone>
    <zone name="Dining">
      [...]
    </zone>
    [...]
  </room>
</group>
```

Abb 3.14 Räume und Raumzonen

Um eine eindeutige Adressierung der Räume und Raumzonen zu gewährleisten, darf bei keinem der Objekte einer Hierarchieebene ein Name mehrfach benutzt werden. Wie bei einem Verzeichnispfad ergibt sich für jedes Element so eine eindeutige ID als Schlüs-

selattribut<sup>30</sup>. Die Zone mit dem Namen „Dining“ in Abbildung 3.14 erhält die aus den Namen der Gruppe, Raum und Zone zusammengesetzte ID „Apartment 1/Living/Dining“. Mit Räumen wird auf dieselbe Weise verfahren. Die IDs von Gruppen erhalten zusätzlich einen abschließenden Schrägstrich, um den Unterschied zu Räumen oder Zonen zu markieren. Die Gruppe „Apartment 1“ in Abbildung 3.13 erhält auf diese Weise die eindeutige ID „Building A/Apartment 1/“.

## Relationen

Beziehungen zwischen je zwei Gruppen, Räumen oder Zonen werden durch das Element `<relation>` beschrieben. Zur exakten Adressierung jeder Relation dienen die IDs der in Beziehung stehenden Funktionsbereiche (Gruppen, Räume oder Zonen). Dies erfolgt durch die zwei- oder mehrfache Zuweisung des Elements `<entity>`. Zur korrekten Bearbeitung müssen selbstverständlich alle in Beziehung stehenden Funktionsbereiche im Dokument korrekt deklariert sein.

```
<relation>
  <entity>Apartment 1/Living/Dining</entity>
  <entity>Apartment 1/Living/Watching TV</entity>
  [...]
</relation>
```

Abb 3.15 Relationen

## Merkmale

Die besonderen Merkmale von Raumgruppen, Räumen und Raumzonen können in XSAL auf vier unterschiedlichen Skalenniveaus beschrieben werden. Hierzu dient das Element `<characteristic>`. Für jedes Merkmal muss eine Bezeichnung über das Element `<caption>` und das Skalenniveau mit Hilfe des Element `<scaletype>` definiert werden. Zur Beschreibung qualitativer Merkmale kann zwischen Nominal- oder Ordinalskala (*nominal*, *ordinal*) gewählt werden. Das Vokabular der möglichen Kategorien wird über das Element `<vocabulary>` festgelegt. Es wird vereinbart, dass die einzelnen in Frage kommenden Kategorien durch einen Strichpunkt getrennt (Semicolon-Separated), zu einem String aneinandergereiht werden. Zusätzlich kann die Wichtigkeit eines Merkmals über das Element `<priority>` durch einen Wert zwischen 0.0 und 1.0 festgelegt werden. Über das Element `<description>` kann erläuternd eine Kurzbeschreibung des betreffenden Merkmals erfolgen.

```
<characteristic>
  <caption>Figure</caption>
  <description>
    The Figure of a certain room means...
  </description>
  <scaletype>Nominal</scaletype>
  <vocabulary>Triangle;Rectangle;T-Shape;L-Shape</vocabulary>
```

<sup>30</sup> Als Schlüsselattribute werden Attribute mit eindeutig identifizierendem Charakter bezeichnet.

```

</characteristic>

<characteristic>
  <caption>Quality of Orientation (illumination)</caption>
  <description>
    The Quality of Orientation of a certain room means...
  </description>
  <scaletype>Ordinal</scaletype>
  <vocabulary>Low;Middle;High</vocabulary>
</characteristic>

```

Abb 3.16 Qualitative Merkmale

Für quantitative Merkmale kann über das Element `<scaletype>` zwischen Verhältnis- und Absolutskala (*ratio*, *absolute*) gewählt werden. Neben der bereits beschriebenen Tags `<caption>` und `<description>` wird die Maßeinheit durch das Element `<unit>` festgelegt.

```

<characteristic>
  <caption>Area Net</caption>
  <description>
    The Area Net of a certain room means...
  </description>
  <scaletype>Ratio</scaletype>
  <unit>m2</unit>
</characteristic>

<characteristic>
  <caption>Number of Windows</caption>
  <description>
    The Number of windows of a certain room means...
  </description>
  <scaletype>Absolute</scaletype>
  <unit>pcs</unit>
</characteristic>

```

Abb 3.17 Quantitative Merkmale

## Präferenzfunktionen

Die Wertefunktion, welche die Präferenzen bezüglich der Ausprägungen eines Merkmals beschreibt, kann durch eine endliche Menge von Punkten beschrieben werden. Dazu werden die einzelnen Koordinaten der Funktion durch die mehrfache Deklaration des Elements `<vertex>` innerhalb des Elements `<valuefunction>` definiert. Die Abbildungen 3.18 bis 3.21 zeigen Beispiele XSAL-notierter Präferenzfunktionen und die dazugehörige grafische Darstellungen. Die Wertefunktion eines qualitativen Merkmals definiert für jede mögliche Merkmalsausprägung beziehungsweise Kategorie genau einen Präferenzwert. Bei quantitativen Merkmalen (Abb. 3.20-21) bilden die einzelnen Präferenzwerte ein Polygon, welches für Datenverarbeitungen zum Beispiel durch Bézier-Kurven interpoliert werden können.<sup>31</sup>

<sup>31</sup> Eine noch präzisere Definition von Präferenzfunktionen könnte ähnlich dem `<path>` element in SVG (Scalable Vector Graphics, ein XML Standard zur Beschreibung zweidimensionaler Vektorgrafik) über die Spezifikation einer Serie miteinander verbundener Linien, Kreisbogen und Kurven erfolgen.

```

<valuefunction>
  <vertex>0,1.0</vertex>
  <vertex>1,0.3</vertex>
  <vertex>2,0.0</vertex>
</valuefunction>

```

Abb 3.18 Qualitative Präferenzfunktion

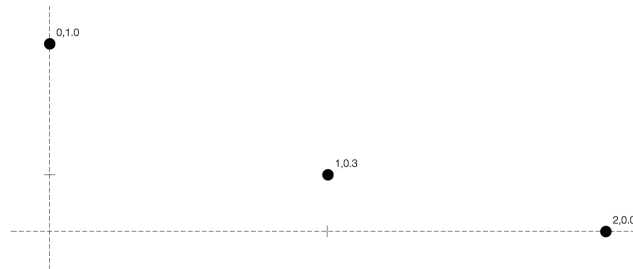


Abb 3.19 Qualitative (diskrete) Präferenzfunktion

```

<valuefunction>
  <vertex>25.0,0.0</vertex>
  <vertex>17.5,0.1</vertex>
  <vertex>29.5,0.3</vertex>
  <vertex>30.5,1.0</vertex>
  <vertex>30.5,0.3</vertex>
  <vertex>32.5,0.1</vertex>
  <vertex>35.0,0.0</vertex>
</valuefunction>

```

Abb 3.20 Quantitative Präferenzfunktion

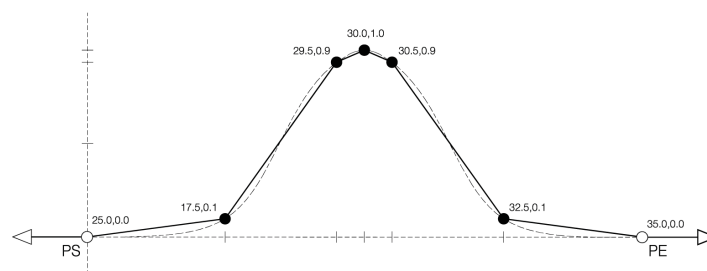


Abb 3.21 Quantitative (stetige) Präferenzfunktion

## Nebenvereinbarungen

Zur maschinellen Verarbeitung von Präferenzfunktionen werden im XSAL Standard Nebenvereinbarungen getroffen, welche festsetzen, wie bestimmte Besonderheiten der Funktionsverläufe bei der Datenverarbeitung interpretiert werden sollen:

1. Durch die Definition von Punkten eines Polygons wird nur ein relevanter Ausschnitt des Merkmalskontinuums beschrieben. Es wird vereinbart, dass Start- und Endpunkt PS und PE für absolut- oder verhältnisskalierte Merkmale eine theoretische Fortsetzung ins Unendliche in Form eines Strahls finden (vgl. Kapitel 5.2).

2. Als y-Koordinaten der Präferenzfunktionen sind grundsätzlich nur Werte kleiner gleich 1.0 zulässig. Größere Werte werden auf 1.0 heruntergesetzt.
3. Negative y-Koordinaten, also Werte der Präferenzfunktion kleiner 0.0 werden ebenfalls gekappt, das heißt auf 0.0 gesetzt.<sup>32</sup>
4. Bei einer unstetigen Entwicklung der Präferenzfunktion (Abb. 3.22) ist jener Punkt mit dem höchsten Wert entscheidend. Diese Vereinbarung ist auch Hilfreich bei der späteren Auswertung von Wertfunktionen, deren Verlauf aufgrund fehlerhafter Modellierung Überhänge oder Selbstüberschneidungen aufweist.

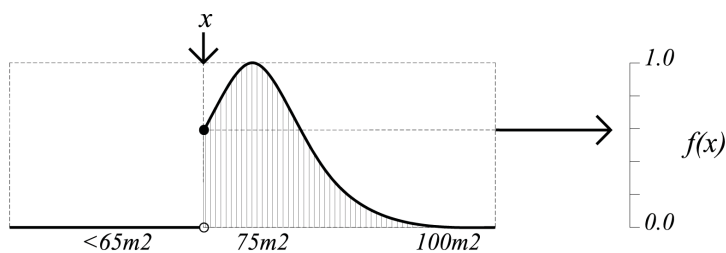


Abb 3.22 unstetige Präferenzfunktion

## Beispiel

Abbildung 3.23 zeigt die einzelnen Elemente der EXTENSIBLE SPACE ALLOCATION LANGUAGE im Zusammenspiel. Konkret handelt es sich bei dem Beispiel um die Spezifikation einer Wohnung, bestehend aus einem  $30m^2$  großen Wohnzimmer mit Esstisch und Fernsehbereich, einer  $12m^2$  großen Küche, einem  $10m^2$  großen Schlafzimmer und einem Sanitärbereich, für den keine bestimmte Größe definiert wurde. Für das Wohnzimmer wird außerdem eine direkte Türverbindung zur Küche gefordert.

```
<spaceallocation>
  <group name="Apartment">
    <room name="Bedroom">
      <characteristic>
        <caption>Area Net</caption>
        <scaletype>Ratio</scaletype>
        <unit>m2</unit>
        <valuefunction>
          <vertex>9.0,0.0</vertex>
          <vertex>10.0,1.0</vertex>
          <vertex>11.0,1.0</vertex>
          <vertex>12.0,0.0</vertex>
        </valuefunction>
      </characteristic>
    </room>
  </group>
</spaceallocation>
```

<sup>32</sup> Es wäre möglich, negative Bereiche einer Präferenzfunktion auch gezielt einzusetzen. Auf diese Weise könnten zum Beispiel K.O.-Kriterien (inakzeptable Merkmalsausprägungen) modelliert werden, welche den Lösungsraum mit *Sperrgebieten* (vgl. Gerdes u.a. 2004, 76.) durchziehen, bei deren ‚Betreten‘ nicht nur der Mindestwert 0.0 vergeben wird, sondern ein zusätzliches *Malus- oder Warnsystem* in Kraft tritt.

```

<room name="Kitchen">
  <characteristic>
    <caption>Area Net</caption>
    <scaletype>Ratio</scaletype>
    <unit>m2</unit>
    <valuefunction>
      <vertex>6.0,0.0</vertex>
      <vertex>7.0,1.0</vertex>
      <vertex>8.0,1.0</vertex>
      <vertex>9.0,0.0</vertex>
    </valuefunction>
  </characteristic>
</room>

<room name="Bathroom & Toilet"></room>

<room name="Livingroom">
  <characteristic>
    <caption>Area Net</caption>
    <scaletype>Ratio</scaletype>
    <unit>m2</unit>
    <valuefunction>
      <vertex>25.0,0.0</vertex>
      <vertex>30.0,1.0</vertex>
      <vertex>35.0,0.0</vertex>
    </valuefunction>
  </characteristic>
  <zone name="Watching TV">
    <characteristic>
      <caption>Area Net</caption>
      <scaletype>Ratio</scaletype>
      <unit>m2</unit>
      <valuefunction>
        <vertex>8.0,0.0</vertex>
        <vertex>9.0,1.0</vertex>
        <vertex>11.0,1.0</vertex>
        <vertex>12.0,0.0</vertex>
      </valuefunction>
    </characteristic>
  </zone>
</room>
</group>
[...]
<relation>
  <entity>Apartment/Livingroom</entity>
  <entity>Apartment/Kitchen</entity>
  <characteristic>
    <caption>Access</caption>
    <scaletype>Ordinal</scaletype>
    <vocabulary>Separated;Doorway;Indirect Access</vocabulary>
    <valuefunction>
      <vertex>0,0.0</vertex>
      <vertex>1,1.0</vertex>
      <vertex>2,0.2</vertex>
    </valuefunction>
  </characteristic>
</relation>
</spaceallocation>

```

Abb 3.23 Gebäudespezifikation in XSAL-Notation

## Dateneingabe

Durch den vorgeschlagenen XSDL Standard eröffnet sich die Möglichkeit, Gebäudespezifikationen in einem beliebigen XML Editor zu erstellen. Hierbei kann auf eine große Anzahl freier sowie kommerzieller Produkte, wie zum Beispiel XML NOTEPAD<sup>33</sup> oder OXYGEN<sup>34</sup> zurückgegriffen werden. Wenn ein XML Schema vorliegt, sind die meisten XML Editoren dazu in der Lage, XSDL Dokumente auf ihre *Gültigkeit* und *Wohlgeformtheit* zu untersuchen. Die Dateneingabe erfolgt bei üblichen Editoren in der Regel über Text-Eingabemasken. Die Verzeichnisstruktur von XSDL Dokumenten kann auch ohne Editor in jedem Standard Webbrowser betrachtet werden.

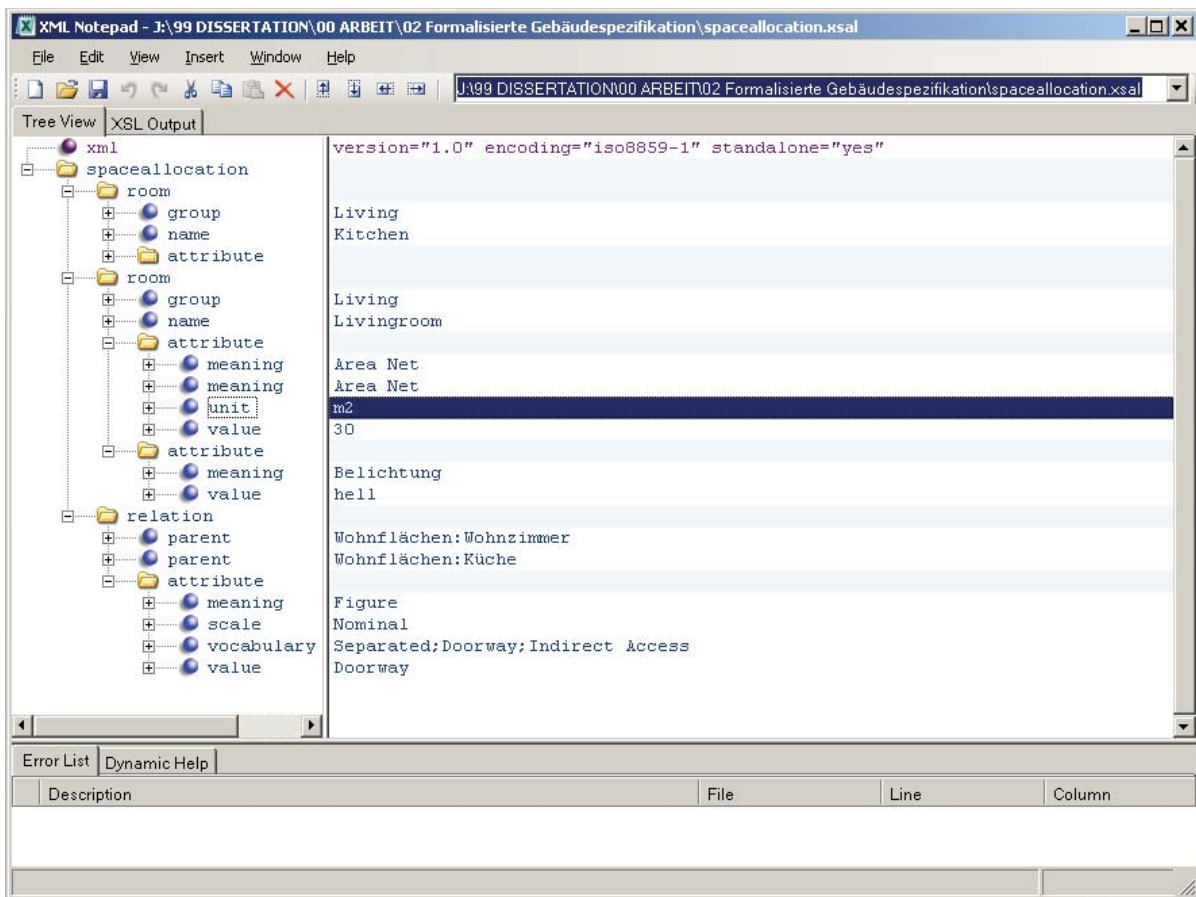


Abb 3.24 XML NOTEPAD

## Datenausgabe

Die Trennung von Struktur, Inhalt und Form gehört zu den Grundprinzipien von Informationstechnologie und Datenmodellierung. XML stellt ein allgemein anerkanntes Beispiel für diese Philosophie dar. Sobald die im Zuge der Bedarfserhebung gesammelten Anforderungen an ein zukünftiges Gebäude einmal durch ein geeignetes Schema hierarchisch strukturiert sind, kann die Darstellung der Inhalte in beliebiger Form erfolgen. Zu diesem Zweck werden sogenannte Stylesheets eingesetzt. Stylesheets sind in einer Sprache wie CSS oder XSL verfasste Formatierungsvorschriften, welche festlegen, wie die Darstellung der Daten

<sup>33</sup> Freier Download auf [www.microsoft.com](http://www.microsoft.com)

<sup>34</sup> Syncro Soft: [www.oxygenxml.com](http://www.oxygenxml.com)

schlussendlich zu erfolgen hat.<sup>35</sup> Mit einem geeigneten Stylesheet ist es möglich, die in XSAL notierten Gebäudespezifikationen in tabellarische Form zu bringen, wie man sie aus den in der Praxis üblichen Raumprogrammen kennt. Genauso können Relationen als Matrix ausgegeben werden (vgl. Kapitel 1.3). Darüber hinaus entsteht durch XSAL die Möglichkeit neue, wesentlich anschaulichere, interaktive und grafische Repräsentationen zu entwickeln. Gebäudespezifikationen können bei einer größeren Anzahl von Funktionsbereichen und Relationen rasch eine hohe Komplexität erlangen. Die Betrachtung vernetzter Datensätze als hierarchische Verzeichnisstruktur verliert dabei zunehmend an Übersichtlichkeit. Die Visualisierung von komplexen Datenstrukturen hat sich in den letzten Jahren zu einer besonderen Sparte des Informationsdesigns entwickelt. Häufig wird dabei auf Strategien der Graphentheorie zurückgegriffen. Abbildung 3.25 zeigt ein Beispiel aus dem Aufsatz *Visualizing Multiple Evolution Metrics* von Pinzger u.a. aus dem Jahre 2005.<sup>36</sup>

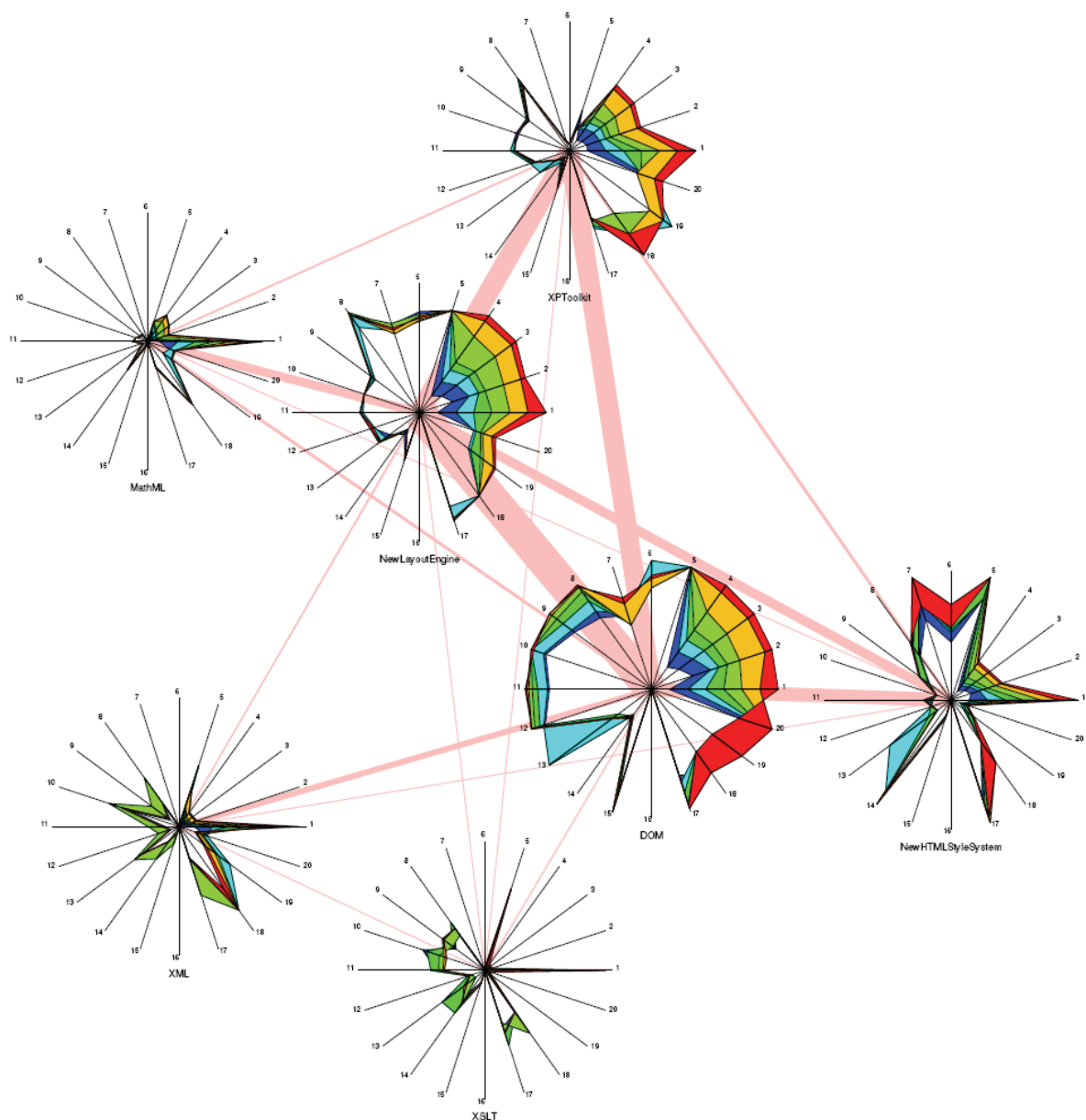


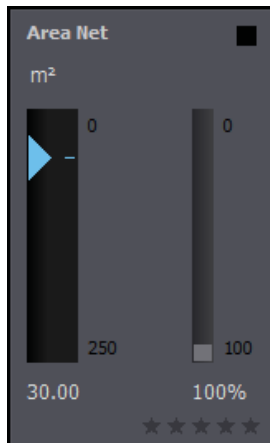
Abb 3.25 Kiviats Graph

<sup>35</sup> Meyer 2005, Tidwell 2002.

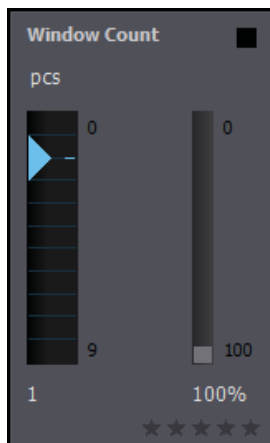
<sup>36</sup> Vgl. Pinzger u.a. 2005, 7.



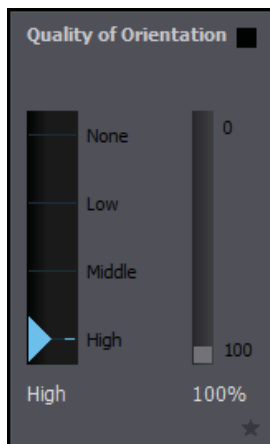




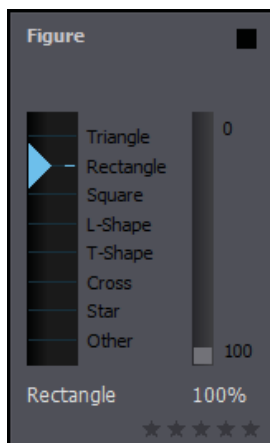
```
<characteristic>
  <caption>Area Net</caption>
  <description>
    [...]
  </description>
  <scaletype>Ratio</scaletype>
  <unit>m²</unit>
  <valuefunction>
    <vertex>15.0,0.0</vertex>
    <vertex>30.0,1.0</vertex>
    <vertex>45.0,0.0</vertex>
  </valuefunction>
</characteristic>
```



```
<characteristic>
  <caption>Window Count</caption>
  <description>
    [...]
  </description>
  <scaletype>Absolute</scaletype>
  <unit>pcs</unit>
  <valuefunction>
    <vertex>1,0.0</vertex>
    <vertex>2,1.0</vertex>
    <vertex>3,0.0</vertex>
  </valuefunction>
</characteristic>
```



```
<characteristic>
  <caption>Quality of Orientation</caption>
  <description>
    [...]
  </description>
  <scaletype>Ordinal</scaletype>
  <vocabulary>None;Low;Middle;High</vocabulary>
  <valuefunction>
    <vertex>0,0.0</vertex>
    <vertex>1,0.3</vertex>
    <vertex>2,0.6</vertex>
    <vertex>3,1.0</vertex>
  </valuefunction>
</characteristic>
```



```
<characteristic>
  <caption>Figure</caption>
  <description>
    [...]
  </description>
  <scaletype>Nominal</scaletype>
  <vocabulary>Triangle;Rectangle;L-Shape;T-Shape</vocabulary>
  <valuefunction>
    <vertex>0,0.0</vertex>
    <vertex>1,1.0</vertex>
    <vertex>2,0.0</vertex>
    <vertex>3,0.0</vertex>
  </valuefunction>
</characteristic>
```

Abb 3.27 Slider und XSAL-Tag für Merkmale unterschiedlicher Skalenniveaus



Abb 3.28 Screenshot CURVE 2, Software Synthesizer mit Waveform Editor (c) www.cableguys.de

Eine noch viel differenziertere grafische Modellierung von Präferenzfunktionen könnte durch die Entwicklung spezieller Eingabemodule erfolgen, welche mehr Gestaltungsmöglichkeiten als einfache Slider bieten. Kurveneditoren werden beispielsweise in Bildbearbeitungsprogrammen zur Definition des Verlaufes von Helligkeit und Kontrast von Rastergrafiken oder in 3D Animationsprogrammen zur Kontrolle von Bewegungsabläufen eingesetzt. Auch bei Software Synthesizern finden sich hochentwickelte und grafisch ansprechend gestaltete Wave Form Editoren (Abb 3.28).

### 3.6 Schlussfolgerungen

Die EXTENSIBLE SPACE ALLOCATION LANGUAGE V1.0 stellt das Grundgerüst eines Standards zur Erstellung differenzierter Gebäudespezifikationen auf Basis der Auszeichnungssprache XML dar. XSAL Dokumente enthalten Informationen über Raumgruppen, abgeschlossene Räume, Zonen innerhalb von Räumen und geben Auskunft über Beziehungen zwischen diesen Bereichen. Durch die genaue Spezifikation der zur Beschreibung von Gebäuden verwendeten Attribute wird die Bedeutung jedes Gebäudemerkmals definiert und dessen Wertemenge möglicher Ausprägungen festgelegt. Durch die Definition von Wertefunktionen können die Zielvorstellungen bezüglich interessierender Eigenschaften, das heißt der Präferenzen eines Entscheiders präzise definiert werden.

Gebäudespezifikationen in XSAL Notation können nicht nur von Menschen gelesen werden, sondern enthalten auch die notwendigen Anweisungen für eine maschinelle Weiterverarbeitung der kommunizierten Inhalte. Da XML keine Programmiersprache ist, enthalten auch XSAL-Dokumente dabei noch keine Elemente für die Steuerung von Programmabläufen. Die im Zuge dieses Kapitels vorgestellten Sprachelemente (Tags) stellen eine einfache, jedoch vielseitige Basis dar. Ergänzungen des Schemas zu einem späteren Zeitpunkt sind durchaus denkbar. Neben der hohen Übersichtlichkeit liegt ein weiterer Hauptvorteil der XML Notation in der einfachen Erweiterbarkeit. Zur Dateneingabe kann jeder beliebige Standard XML Editor benutzt werden. Die Datenausgabe erfolgt mit Hilfe von Stylesheets in beliebiger Form, als Tabelle, Matrix, Grafik etc.. Durch die Entwicklung von speziellen XSAL-Editoren ist es möglich, die Daten Ein- und Ausgabe interaktiv zu gestalten. Durch die Entwicklung von geeigneten grafischen User Interfaces kann, wie am Beispiel des DESIGN GOAL MANAGERS gezeigt wurde, eine sehr komfortable Eingabe und Organisation der komplexen Datensätze einer Gebäudespezifikation erfolgen.

## 4 BUILDING INFORMATION MODEL

*»Essentially, all models are wrong, but some are useful«<sup>1</sup>*

George Box 1987

Erst vor ein paar Jahren hat Peter Norvig, der Leiter der Forschungsabteilung von Google, diesem Ausspruch des Statistikers George Box<sup>2</sup> ein Update verpasst und postuliert, dass alle Modelle tatsächlich falsch wären, aber dass man auch immer häufiger ohne sie Erfolg haben kann. Firmen wie Google sind nach Norvig im Zeitalter riesiger, frei zugänglicher, aktueller Information nicht weiter auf fehlerhafte abstrakte Modelle und Simulationen angewiesen. Wenn man genügend Daten besitzt und über geeignete mathematische Algorithmen zu deren Analyse verfügt, dann sprechen diese in dieser Betrachtungsweise ohnehin für sich.<sup>3</sup> Schon Nelson Goodman hat darauf hingewiesen, dass nur wenige Ausdrücke im populären und wissenschaftlichen Diskurs undifferenzierter gebraucht werden als der des Modells. Nach Goodman können durchaus auch Data-Mining Algorithmen als Modelle bezeichnet werden.

*»Ein Modell ist etwas, das man bewundert oder dem man nacheifert, ein Muster, ein passender Fall, ein Typ, ein Prototyp, ein Exemplar, ein Modell in Originalgröße, eine Mathematische Beschreibung - nahezu alles von einer nackten Blondine bis zu einer quadratischen Gleichung -, und das zu dem, wofür es „Modell ist“, in fast jeder Symbolisierungsrelation stehen kann.«<sup>4</sup>*

Für die Architektur sind Modelle in jedem Fall nach wie vor von großer Bedeutung. Der Begriff *Modell* entstand in der Renaissance aus dem italienischen Wort *modello*, der Bezeichnung für eine vorbereitende Modellstudie eines Bau- oder Kunstwerks in verkleinertem Maßstab (lat. *modulus*).<sup>5</sup> Seit Jahrhunderten werden Architekturmodelle angefertigt, um Qualitäten von Gebäudeentwürfen zu überprüfen. Das analoge, physische Architekturmodell stellt ein unersetzliches Arbeitsmittel im architektonischen Entwurfsprozess dar. Seit dem Einsatz von Computern spielen auch digitale Gebäudemodelle eine immer wichtigere Rolle. Die Verlagerung der Architekturproduktion auf den Computer findet heute den Höhepunkt im sogenannten *Building Information Modeling* (BIM).

Mit BIM hat nun auch die Architektur das erreicht, was sich in der Fahrzeugindustrie oder im Anlagenbau schon längst unter dem Begriff *Product Lifecycle Management* etabliert hat. Sämtliche planungs-, ausführungs- und nutzungsrelevante Gebäudedaten werden in einem zentralen Datenmodell vereinigt, welches ein Gebäude über seinen gesamten Lebenszyklus

---

<sup>1</sup> Box/Draper 1987, 424.

<sup>2</sup> George Edward Pelham Box war ein britischer Statistiker und emeritierter Professor der University of Wisconsin, Madison

<sup>3</sup> Vgl. Anderson 2013, 124.

<sup>4</sup> Goodman 2012, 164.

<sup>5</sup> Vgl. Wikipedia en., *Modello*, 23.03.2013.

hinweg begleiten soll - von der Konzeption bis zu seiner Entsorgung oder Umnutzung. Alle Projektbeteiligten aus Architektur- und Fachplanung, ausführende Gewerke sowie Nutzerinnen und Nutzer greifen auf diese gemeinsame Informationsbasis (Datenbank) zu, welche im Zuge des Projektfortschritts kollaborativ immer weiter verfeinert wird. Sämtliche Pläne wie Grundrisse, Schnitte oder Ansichten, aber auch Perspektiven oder Bauteillisten stellen in der Folge nur noch unterschiedliche Ansichten ein und desselben Datensatzes dar. Das Wegfallen fehlerträchtiger Mehrfacheingaben der Gebäudegeometrie in die jeweilige Spezialsoftware der Fachplaner (z.B. aus den Bereichen Statik oder Bauphysik) oder das Vermeiden von zeitraubenden Aktualisierungsarbeiten des Plansatzes nach Änderungen, verspricht eine Steigerung der Konsistenz und Qualität der Planung. Das US-amerikanische *National Building Information Model Standard Project Committee* definiert BIM mit folgenden Worten:

*»Building Information Modeling (BIM) is a digital representation of physical and functional characteristics of a facility. A BIM is a shared knowledge resource for information about a facility forming a reliable basis for decisions during its life-cycle; defined as existing from earliest conception to demolition.«<sup>6</sup>*

Das unter Softwarearchitektinnen und -architekten verbreitete *Don't Repeat Yourself!* (DRY), welches darauf abzielt, bei der Programmierung Redundanzen zu vermeiden, soll nun wohl auch für Baukünstlerinnen und Baukünstler gelten. Der Einzug des *Building Information Modelings* markiert einen Paradigmenwechsel in der Geschichte des *Computer Aided Architectural Designs* (CAAD). Im folgenden Kapitel erfolgt eine kurze Darstellung der historischen Entwicklung von BIM. Danach wird das Programm DRAFTPAD vorgestellt. Dieses vom Autor entwickelte Zeichenprogramm basiert auf einem simplen assoziativen Gebäudemodell, auf welches die in dieser Arbeit vorgenommenen Untersuchungen aufsetzen werden. Die Beschreibung der einzelnen Entitäten dieses Modells wird durch relevante Teile des Quelltextes der Typelibrary begleitet.

## 4.1 Historische Entwicklung

Die Ursprünge des *Building Information Modeling* gehen auf die frühen 60er Jahren des 20. Jahrhunderts zurück. Bereits im Jahre 1962 lieferte Douglas Engelbart die beeindruckend vorausschauende Vision eines „*augmented architect*“<sup>7</sup>. Der Computertechniker und Erfinder hatte eine beinahe unheimlich klare Vorstellung davon, wie die Arbeitsumgebung von Architektinnen und Architekten durch den Einsatz von objektorientiertem, parametrischen Design und relationalen Datenbanken in Zukunft durch den Einsatz von Computern aussehen könnte.

*»Let us consider an augmented architect at work. He sits at a working station that has a visual display screen ... this is his working surface, and is controlled by a computer (his "clerk") [...] He is designing a building. He has already dreamed up several basic layouts and structural forms, and is trying them out on the screen. [...] The surveying data for the layout he is working on now have*

<sup>6</sup> [www.nationalbimstandard.org](http://www.nationalbimstandard.org); 22.07.2014.

<sup>7</sup> Engelbart 1962, 5.

*already been entered, and he has just coaxed the clerk to show him a perspective view of the steep hillside building site with the roadway above [...] After a moment, the architect changes the scene on the screen to an overhead plan view of the site [...] the architect next begins to enter a series of specifications and data – a six-inch slab floor, twelve-inch concrete walls eight feet high within the excavation, and so on. When he has finished, the revised scene appears on the screen. A structure is taking shape. He examines it, adjusts it, pauses long enough to ask for handbook or catalog information from the clerk at various points, and readjust accordingly. He often recalls from the "clerk" his working lists of specifications and considerations to refer to them, modify them, or add to them. These lists grow into an evermore-detailed, interlinked structure, which represents the maturing thought behind the actual design.»<sup>8</sup>*

Die 60er und 70er Jahre des 20. Jahrhunderts waren von einem starken Glauben und höchsten Erwartungen an die Computertechnologie geprägt. In dieser Zeit wurden auch die Grundsteine (heutiger) moderner BIM Plattformen gelegt. Edgar Codd entwickelte Anfang der 70er Jahre am *IBM Almaden Research Center* die ersten Grundlagen für experimentelle relationale Datenbanken. Ivan Sutherland sorgte mit seinem Programm *Sketchpad* im Jahre 1962 für einen Meilenstein in der Entwicklung grafischer User-Interfaces. In der Flugzeugindustrie (z.B. Lockheed Flugzeugbau USA) wurden auf IBM Großrechnern die ersten Anläufe im kommerziellen Einsatz von CAD Systemen gestartet. Schon in dieser Zeit, vor der Verfügbarkeit von Personal Computern, beflügelten Architekten wie Herbert Simon, Nicholas Negroponte oder Christoph Alexander die Entwicklung des Computer Aided Designs durch visionäre methodische Ansätze. Als erster Vorreiter heutiger BIM Systeme gilt das *Building Description System* (BDS)<sup>9</sup> von Charles „Chuck“ Eastman aus dem Jahre 1974. Eastmans Computerprogramm bestand aus der erstmaligen Verbindung einer Datenbank mit einem grafischen User-Interface (GUI). In dem Paper *The Use of Computers instead of Drawings in Building Design* formulierte Eastman schon im Jahre 1975 die wesentlichen Argumente für die Verwendung eines zentralen Datenmodells bei der Planung von Gebäuden, welche noch heute wichtige Grundprinzipien der allgemeinen Philosophie von BIM darstellen.

*»It would combine the positive aspects of both drawings and models and eliminate their common weakness. It would incorporate three-dimensional information in an easy-to-read format and would require any change to be made only once for its full effects to be revealed.«<sup>10</sup>*

Eastman versprach durch den Einsatz des Building Description Systems Planungskosten für ein Gebäude um mehr als 50% senken zu können. Dafür führte er zwei wesentliche Gründe an: Der erste Vorteil von BDS gegenüber den herkömmlichen Planungsmethoden lag laut Eastman in einer Effizienzsteigerung durch das Vermeiden von Redundanzen und die Möglichkeit, in sehr kurzer Zeit jeden gewünschten Plan aus einem einzigen stets aktualisierten Datensatz generieren (ableiten) zu können. Als zweiten Vorteil führte Eastman die Möglichkeit an, innerhalb des zentralen Datenmodells verschiedenste Analysen durchfüh-

---

<sup>8</sup> Engelbart 1962, 5.

<sup>9</sup> Eastman 1975.

<sup>10</sup> Eastman 1975, 1.

ren zu können, um die Performance von Entwurfslösungen maschinell evaluieren zu können, ohne davor eine aufwendige Aufbereitung der Daten vornehmen zu müssen, wie es bei herkömmlichen analogen Plänen unvermeidbar der Fall ist.<sup>11</sup>

*»because the building description is now in a machine readable form, any type of quantitative analysis could be directly, coupled to the system. All data preparation for such analyses would be automatic, greatly reducing their cost.«<sup>12</sup>*

Im Laufe der späten 1970er und frühen 1980er Jahre wurde die Weiterentwicklung Eastmans experimentellen Prototyps in Europa vor allem in England vorangetrieben und die ersten Bemühungen einer Kommerzialisierung dieser Technologie unternommen. Während die Methode in den USA als *Building Product Model* beworben wurde, benutzte man in den 1980er Jahren in Europa eher den Ausdruck *Product Information Model*. Daraus entwickelte sich im Laufe der Zeit der heute international gebräuchliche Ausdruck *Building Information Modeling* oder kurz BIM.<sup>13</sup>

Einen wichtigen Beitrag zur Entwicklung der heutigen BIM-Standards leisteten neben zahlreichen anderen der ungarische Physiker Gábor Bojár sowie die Softwareentwickler Leonid Raiz und Irwin Jungreis. Von Bojár, dem späteren Mitbegründer der Firma *Graphisoft*, stammen die ersten Codezeilen des im Jahre 1982 entwickelten Programms *Archicad*, welches die erste auf einem Apple Personal Computer verfügbare BIM Software darstellt.<sup>14</sup> Raiz und Jungreis veröffentlichten im Jahre 2000 mit ihrer damaligen Firma Revit Technology Corporation die erste Version des Programms Revit unter Windows, welches zwei Jahre später von der Firma Autodesk übernommen wurde. Andere architekturenspezifische BIM Plattformen sind beispielsweise das seit dem Jahre 1984 erhältliche *Allplan* von *Nemetschek*, die im Jahre 2003 entwickelte Software *Generative Components* von *Bentley Systems*, oder *Digital Project* von *Gehry Technologies* aus dem Jahre 2004. Zur Gewährleistung der Kompatibilität der unterschiedlichen BIM Systeme wurde Ende der 1990er Jahre der IFC Standard<sup>15</sup> (Industry Foundation Classes) zum Zweck eines Plattformübergreifenden Datenaustausches entwickelt, welcher weltweit durch die non-profit Organisation buildingSMART propagiert wird.<sup>16</sup> Der IFC Standard basiert ursprünglich auf der Datenmodellierungssprache EXPRESS. Seit IFC2x existiert auch eine konsistente XML Spezifikation<sup>17</sup> (IFC-XML) in Form eines XML Schema 1.0 gemäß den Definitionen des W3C<sup>18</sup>.

In der Regel sind die Datenmodelle, welche den meisten etablierten BIM Plattformen zugrunde liegen, sehr komplex aufgebaut und verfügen über eine große Anzahl assoziativ miteinander verknüpfter Datenobjekte. Das ist ein Resultat des ganzheitlichen Ansatzes von BIM. *Building Information Modelle* stellen den bewussten Versuch einer 1:1-Abbildung von realen Gebäuden dar. Das Ziel besteht darin, sämtliche Informationen über ein Gebäude informationstechnisch zu erfassen.

<sup>11</sup> Vgl. Eastman u.a. 1974, 6.

<sup>12</sup> Eastman u.a. 1974, 6.

<sup>13</sup> Vgl. Eastman u.a. 2008, XI.

<sup>14</sup> Vgl. Quirk 2012.

<sup>15</sup> Vgl. ISO 16739.

<sup>16</sup> [www.buildingsmart.org](http://www.buildingsmart.org)

<sup>17</sup> Vgl. ISO10303-28.

<sup>18</sup> World Wide Web Consortium zur Entwicklung offener Internet Standards.



## Wissenschaftliche Modelle – Keep it simple, stupid!<sup>19</sup>

Bei der Entwicklung von Modellen, welche für empirische Untersuchungen herangezogen werden, gelten im Allgemeinen andere Grundsätze. Das Wesen wissenschaftlicher Modelle liegt in der Reduktion der Komplexität des Modells gegenüber der Realität. Nach der *Allgemeinen Modelltheorie* des Philosophen Herbert Stachowiak sollten Modelle keinesfalls sämtliche Attribute des Originals erfassen, sondern nur jene, die für die ModelliererIn oder den Modellierer beziehungsweise die Nutzerin oder den Nutzer des Modells relevant sind. Ein Modell für wissenschaftliche Untersuchungen sollte an Komplexität hinter der Realität zurück bleiben, damit es die Wirklichkeit an Verständlichkeit übertreffen kann.<sup>20</sup> Eine der wichtigsten Eigenschaften jedes guten Modells ist die *Isomorphie* zur Realität. Im Falle von Gebäudeentwürfen handelt es sich um eine zukünftige Realität. Durch die Analyse des Modells müssen sich Rückschlüsse auf die Qualitäten des modellierten Gebäudes ziehen lassen.<sup>21</sup> Die Kunst bei der Datenmodellierung beziehungsweise der Erstellung eines geeigneten Modells liegt gerade in der Wahl des geeigneten Abstraktionsniveaus. Um dem ganzheitlichen Anspruch von BIM gerecht zu werden sind die Gebäudedatenmodelle der etablierten BIM Plattformen, aufgrund der ständig steigenden Anforderungen an Bauwerke, heute zu hoch komplexen Datenbanken mit nur schwer überschaubaren Datenstrukturen herangewachsen. Der Detailgrad geht bei den meisten Objekten weit über den eines Wettbewerbsentwurfs hinaus. Für die in der vorliegenden Arbeit vorgenommenen Untersuchungen wird aus diesem Grund ein wesentlich einfacheres Gebäudedatenmodell benutzt, welches vom Autor ursprünglich für das Zeichenprogramm DRAFTPAD entwickelt wurde. Dieses speziell zur Erstellung von Wettbewerbsbeiträgen konzipierte Tool wurde im Jahre 2009 implementiert und wird seit damals im praktischen Einsatz erprobt.

## 4.2 Draftpad

Das Computerprogramm DRAFTPAD wurde zum Skizzieren architektonischer Ideen entwickelt. Das Tool ist sehr einfach zu bedienen und dreidimensionale Gebäudeentwürfe können relativ schnell modelliert werden. DRAFTPAD (Abb 4.1) besteht im Kern aus einem stark vereinfachten *Building Information Model*, welches sich bewußt auf jenen Detailgrad beschränkt, der für die Abbildung eines üblichen Beitrags zu einem Architekturwettbewerb notwendig ist. Die Konturen der gewünschten Räume werden wie Plateaus auf einer oder mehreren Ebenen gezeichnet. Danach können Elemente wie Türen, Fenster oder vertikale Erschließungselemente platziert werden. Zur Möblierung steht eine Sammlung von parametrischen Objekten zur Verfügung, deren spezifische Eigenschaften numerisch festgesetzt werden. Zur Manipulation des räumlichen Arrangements stehen zahlreiche, für diesen Zweck maßgeschneiderte Werkzeuge zur Verfügung. Neben Import- und Exportfunktionen zum Datenaustausch mit anderen CAD Programmen erfolgt die Speicherung des Gebäudedatenmodells in einem nativen Datenformat.

---

<sup>19</sup> KISS (Keep it simple, stupid!) stellt ein im Feld der Softwareentwicklung verbreitetes Prinzip dar, welches besagt, dass eine möglichst einfache Lösung eines Problems gewählt werden sollte.

<sup>20</sup> Vgl. Stachowiak, 131-133.

<sup>21</sup> Vgl. Suhl/Mellouli 2006, 7.



Abb 4.1 Martin Emmerer 2009, DRAFTPAD

## Entitäten

Die Auswahl der Entitäten erfolgte bei der Entwicklung des in DRAFTPAD verwendeten Datenmodells in Anlehnung an das sogenannte Pareto-Prinzip<sup>22</sup>. Die Entwurfspraxis zeigt, dass Gebäudeentwürfe trotz ihrer formalen Vielfalt häufig nur eine beschränkte Zahl wiederkehrender generischer Elemente enthalten. Nur die häufigsten dieser Elemente wurden bei der Entwicklung der Modellstruktur als Entitäten identifiziert, und wiederum nur deren relevantesten Eigenschaften als Attribute in Objektklassen aufgenommen. Die Einfachheit der

<sup>22</sup> Der im 19. Jahrhundert geborene italienische Ingenieur und Ökonom Vilfredo Pareto fand heraus, dass sich sehr viele Aufgaben mit einem Mitteleinsatz von ca. 20% so erledigen, dass 80% aller Probleme gelöst werden können. Die Anwendbarkeit dieses sogenannten Pareto-Prinzips auf den Entwurf von Gebäudedatenmodelle ist nicht belegt.

Datenstruktur führt in der Folge zu gewissen Einschränkungen: Geschosdecke können nur aus horizontalen ebenen Flächen gebildet werden. Räume haben stets einheitliche Höhen. Wände sind immer vertikal. Wandöffnungen können nur durch die Angabe von Breite und Höhe definiert werden, sind also stets rechteckig. Viele Entwürfe von Gebäuden halten sich jedoch ohnehin aus Gründen der Wirtschaftlichkeit oder Nutzbarkeit an genau diese Regeln und lassen sich so in dem vorgestellten Datenmodell ohne wesentliche Informationsverluste abbilden. Abbildung 4.3 zeigt symbolisch die im Modell zur Verfügung stehenden Datenobjekte mit einer kurzen Spezifikation der Attribute, welche zur Beschreibung dieser Objekte herangezogen werden. Abbildung 4.4 zeigt auszugsweise die, für diese Arbeit relevanten Teile des Quellcodes. Mit den Entitäten des Gebäudedatenmodells lassen sich sehr unterschiedliche Raumkonzepte abbilden. Neben Abfolgen geschlossener Räume können beispielsweise auch offene Grundrisse modelliert werden. Abbildung 4.2.a zeigt den in DRAFTPAD nachgebildeten Entwurf eines Golfclubhauses von Mies van der Rohe, welches eine *offenen Grundriss* aufweist. Mit Hilfe der Entität *Zone* ist es möglich innerhalb eines Raumkontinuums räumliche Phänomene, wie Nutzungsüberschneidungen oder Doppelnutzungen abzubilden und in der Folge auch analytisch zu betrachten.

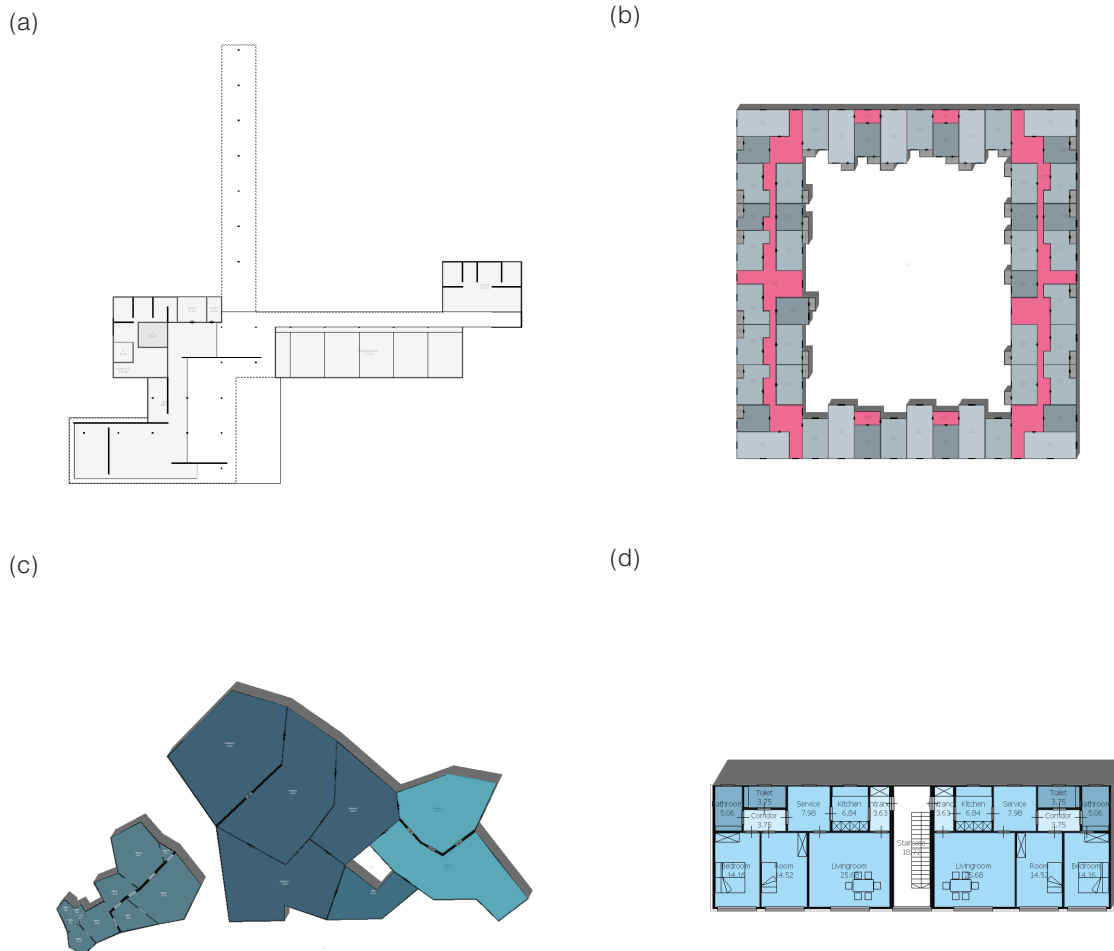
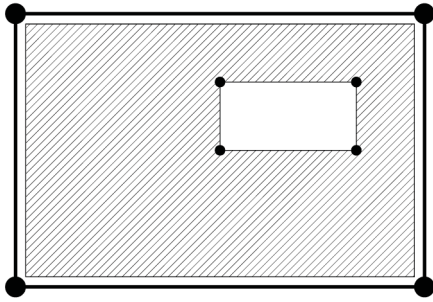


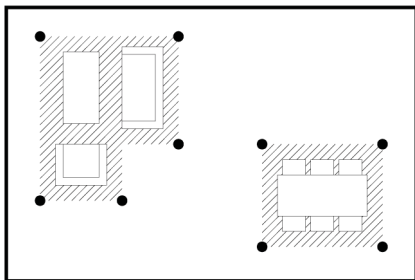
Abb 4.2 Martin Emmerer 2009, Screenshots DRAFTPAD

## Semantik

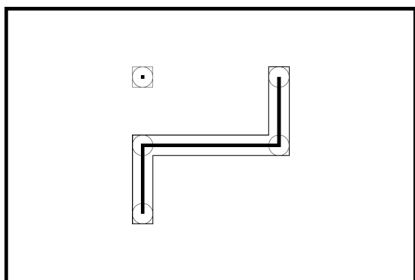


Die Entität ROOM beschreibt einen abgeschlossenen Raum. Die Geometrie wird durch die Brutto- und Netto-raumbegrenzung in Form von zwei einfachen Polygonen und einer Höhenangabe bestimmt. Ein weiterer Parameter bestimmt die Geschosshöhe.

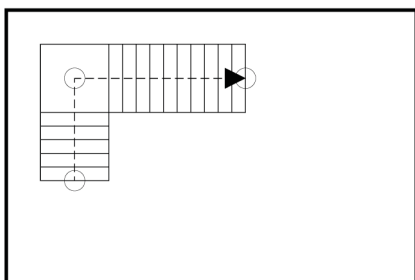
Deckendurchbrüche werden ebenfalls durch Polygone definiert. Jeder Raum ist im Modell über seinen Namen eindeutig identifizierbar.



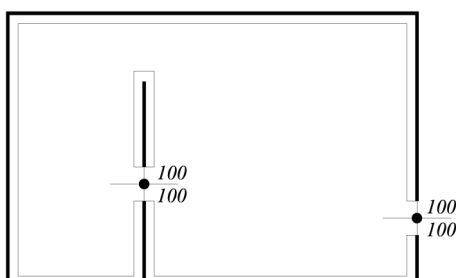
Die Entität ZONE beschreibt einen Bereich innerhalb einer Instanz von ROOM. Die äußere Kontur wird durch ein einfaches Polygon definiert. Jede Region ist im Datenmodell über ihren Namen eindeutig identifizierbar. Jede Region speichert die Information, in welchem Raum sie sich befindet (Parent). Weitere Zeichenelemente werden in einem Block aus geometrischen Elementen abgelegt.



Die Entität WALL beschreibt Wandelemente oder Stützen. Die Geometrie jedes Wandobjekts wird durch eine Konturlinie (offenes Polygon) sowie eine Höhenangabe definiert. Jede Instanz speichert die Information darüber, in welchem Raum sie sich befindet (Parent).



Die Entität STAIR beschreibt vertikale Erschließungselemente wie Treppen oder Rampen. Die Geometrie wird durch ein einfaches Polygon beschrieben, welches die Gehlinie repräsentiert. Der Antrittspunkt jeder Instanz entspricht dem Niveau des Raumes, in dem es abgesetzt wurde. Der Austrittspunkt wird durch eine Höhenangabe definiert. Jede Instanz speichert die Informationen über den An- und Austrittsraum (Linkedrooms).



Die Entität OPENING beschreibt Wandöffnungen wie Türen oder Fenster. Die Beschreibung jeder Öffnung erfolgt durch die Angabe eines Basispunktes, sowie die Parameter Breite, Parapet und Höhe. Jede Wandöffnung speichert die Information über die Räume, welche sie verbindet (Linkedrooms).

Abb 4.3 Entitäten und Attribute

## Datenstruktur

```

TRoom = Object
  Height, Level      :Real;
  Name               :String;
  Poly, NetPoly      :T2DPoly;
  Holes              :array of T2DPoly;
  Index              :integer;
  ADJ_R              :TIndexlist;
  [...]
end;

-----

TZone = Object
  Height             :Real;
  Name               :String;
  Poly               :T2DPoly;
  Body               :TBlock;
  Index              :integer;
  Parent             :Integer;
  [...]
end;

-----

TWall = Object
  Height             :Real;
  Name               :String;
  Poly               :T2DPoly;
  Index              :integer;
  Parent             :Integer;
  [...]
end;

-----

TStair = Object
  Poly               :T2DPoly;
  Width, Height      :Real;
  Index              :integer;
  Linkedrooms        :Array [0..1] of Integer;
  [...]
end;

-----

Type      = (otWindow, otDoor)
TOpening = Object
  OType              :TObjectType;
  Index              :Integer;
  Level, Parapet     :Real;
  Height, Width      :Real;
  BasePoint          :T2DPoint;
  Name               :String;
  Linkedrooms        :Array [0..1] of Integer;
  [...]
end;

```

## Assoziativität

Die Objekte des in DRAFTPAD verwendeten Gebäudedatenmodells sind assoziativ miteinander verknüpft. Jeder Raum trägt die Information über direkt benachbarte Räume in sich (ADJ\_R). Regionen und Wandelemente speichern die Information, in welchem Raum sie sich befinden (PARENT). Wandöffnungen sind darüber informiert, welche zwei Räume sie verbinden, oder welcher Raum die Öffnung in den Außenraum beziehungsweise ein Atrium führt (Linkedrooms). Vertikale Erschließungselemente „kennen“ auf dieselbe Weise die Räume, welche sie verbinden. Aus diesen Informationen können duale Graphen abgeleitet werden. Die Bildung des Adjacency- und des Accessgraphen (vgl. Kapitel 2.3) gehört zu den Grundfunktionen des in DRAFTPAD benutzten Building information Models. Diese Graphen können für die Analyse einer Entwurfslösung herangezogen werden. Abbildung 4.5 zeigt einen automatisch generierten Access- und Adjacencygraph.

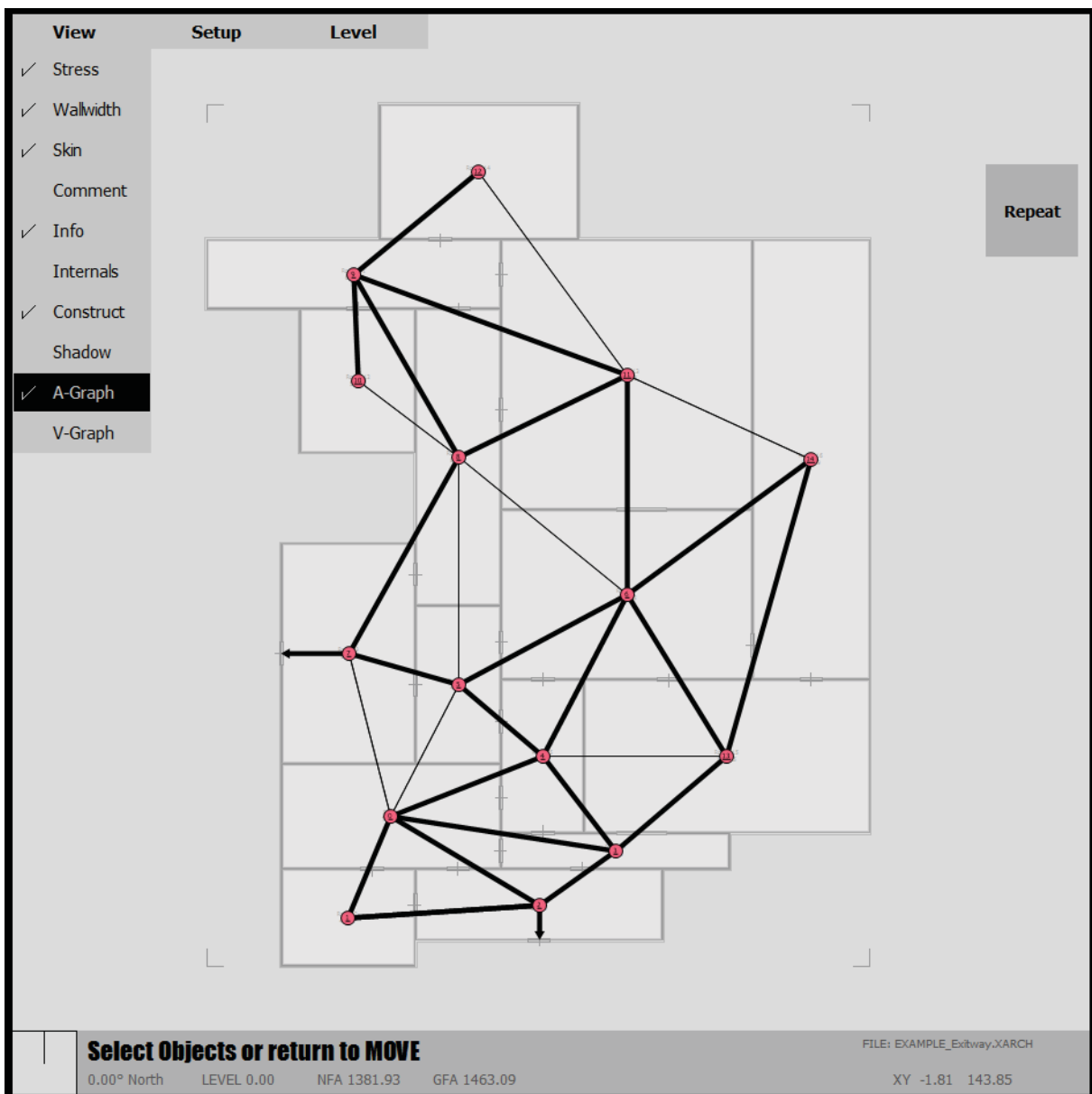


Abb 4.5 Martin Emmerer 2009, Screenshot DRAFTPAD, Adjacency- und Access Graph

## Routing

Durch den Access Graph eines Gebäudes, ist es möglich die Abfolge der bis zum Notausgang zu durchlaufenden Räume zu bestimmen. Für die Berechnung von Wegen, welche Personen innerhalb eines Gebäudes zurücklegen (wie beispielsweise Fluchtwege oder Wege zwischen zwei Räumen) ist diese Graphenstruktur aber noch nicht detailliert genug. Der kürzeste Weg zwischen zwei beliebigen Punkten innerhalb eines Gebäudes verläuft nicht durch die bestehenden Knoten in den Zentren der einzelnen Räume. Auch sind nicht die Raumzentren Ausgangspunkt der Flucht, sondern die vom Eintrittspunkt in den Raum weitest entfernten Raumecken. Aus diesem Grund ist es notwendig, jeden Knoten des Accessgraphen wiederum durch einen Teilgraphen (Subgraph) zu ersetzen, dessen Knoten und Kanten die jeweilige Raumgeometrie abbilden.

Prinzipiell existieren verschiedene Ansätze zur Generierung von Graphen für die Berechnung von Wegen innerhalb eines Gebäudemodells. Die Wahl der geeigneten Methode hängt vom Verwendungszweck ab. Ein solcher Graph kann beispielsweise durch Rasterung (Abb 4.7a) oder Triangulation (Abb 4.7b) der Grundrissgeometrie erzeugt werden.<sup>23</sup> Eine weitere, sehr exakte Möglichkeit besteht in der Berechnung des sogenannten Visibility-Graph (Abb 4.7c). Vor allem wenn es um die Berechnung kürzester Wege (Shortest Path) in einem Architekturmodell geht, zeichnet sich die Anwendung dieser Methode durch deutliche Vorteile gegenüber anderen Konzepten aus. Der Sichtbarkeitsgraph eines Gebäudes wird gebildet durch das Verbinden aller Raum- und Hindernisecken zwischen denen eine direkte Sichtverbindung besteht. Der Visibility Graph ist somit ein vollständiger Graph ohne jene Kanten, welche ein Objekt schneiden.

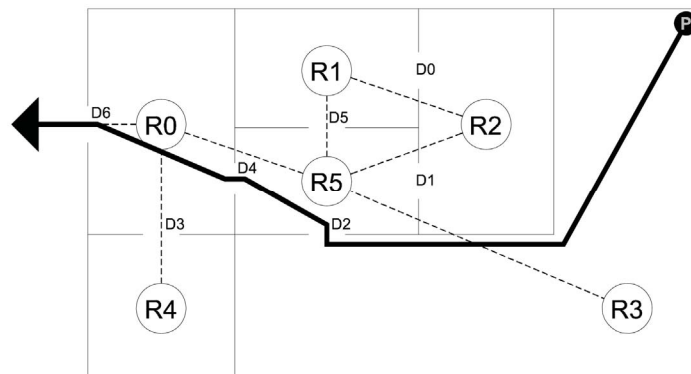
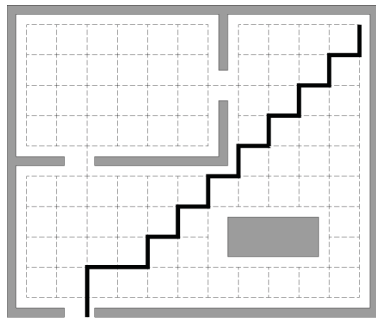
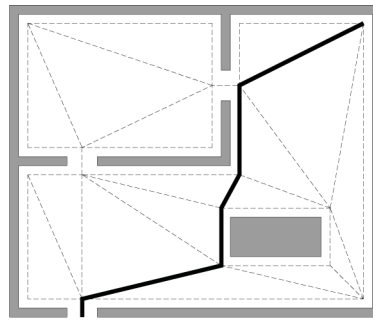


Abb 4.6 Access Graph vs. Fluchtweg aus Raum 3

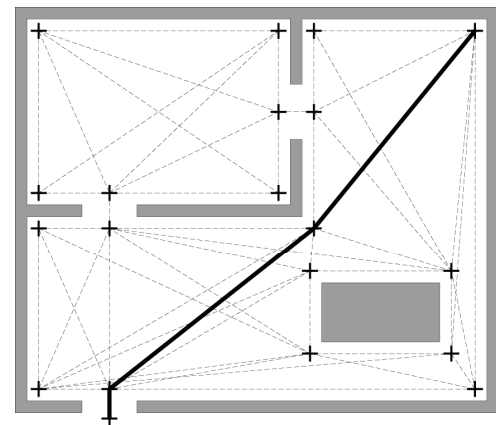
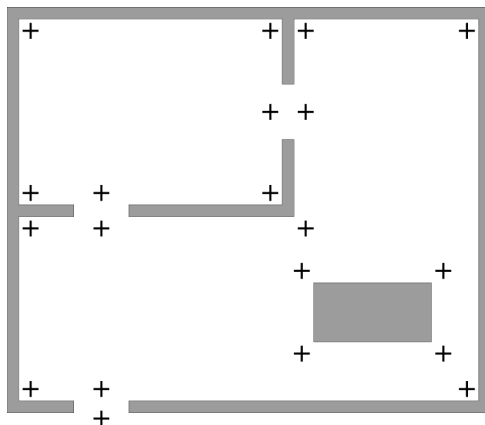
<sup>23</sup> Diese Techniken werden beispielsweise in Videospiele zur Navigation der Charaktere durch künstliche Welten eingesetzt, da auf diese Weise auch bewegte Hindernisse gut modelliert werden können. Innerhalb der Rasterstruktur bleibt der Rechenaufwand unabhängig von der Komplexität der Welt annähernd immer der gleiche. Die Verwendung eines Netzes aus Dreiecken ist in manchen Fällen vorteilhaft, wenn die Grafikkarte dieses zur Darstellung ohnehin schon berechnet hat.



(a) GRID: 168 NODES / 336 EDGES  
**L=19**



(b) MESH: 20 NODES / 39 EDGES  
**L=14.9**



(c) V-GRAPH: 20 NODES / 50 EDGES  
**L=13.8**

Abb 4.7 Verschiedene Methoden zur Bildung kürzester Wege innerhalb eines Polygons  
(a) Rasterung (b) Triangulation (c) Visibility-Graph

Der Visibility-Graph stellt die strengste aller Methoden dar und liefert immer die optimale Wegführung. Ein vollständiger, zweidimensionaler Visibility-Graph enthält immer alle kürzesten Wege zwischen den Knoten.<sup>24</sup> Somit ist eine Optimierung oder Nachbesserung des errechneten Weges nicht mehr notwendig. Ein Nachteil des Visibility-Graph besteht darin, dass bestimmte Grundrissgeometrien zu einer sehr großen Zahl von Kanten führen können. Dies wäre etwa bei runden, durch Polygone abgebildeten Räumen der Fall. Die Rasterung der Fläche kann in solchen Fällen zu wesentlich schnelleren Ergebnissen führen. Die errechneten Wegstrecken sind jedoch beim Rasterverfahren in der Regel deutlich länger, da diese nicht die direkten Wegverbindungen abbilden. Die mit dem Raster- oder Triangulierungs-Verfahren berechneten Wege können somit beispielsweise nicht als feuerpolizeilicher Nachweis der exakten Länge von Fluchtwegen benutzt werden. Eine aufwendige Nachbearbeitung oder Bereinigung suboptimaler Wegführungen geht ebenfalls auf Kosten der Rechenzeit.<sup>25</sup> Abbildung 4.8 zeigt den Visibility-Graph eines beispielhaften Grundrisses.

<sup>24</sup> Vgl. Liu u.a. 1992.

<sup>25</sup> In der Informatik wird eine rechenintensive und somit langsame Programmfunktion als „expensive Function“ bezeichnet.



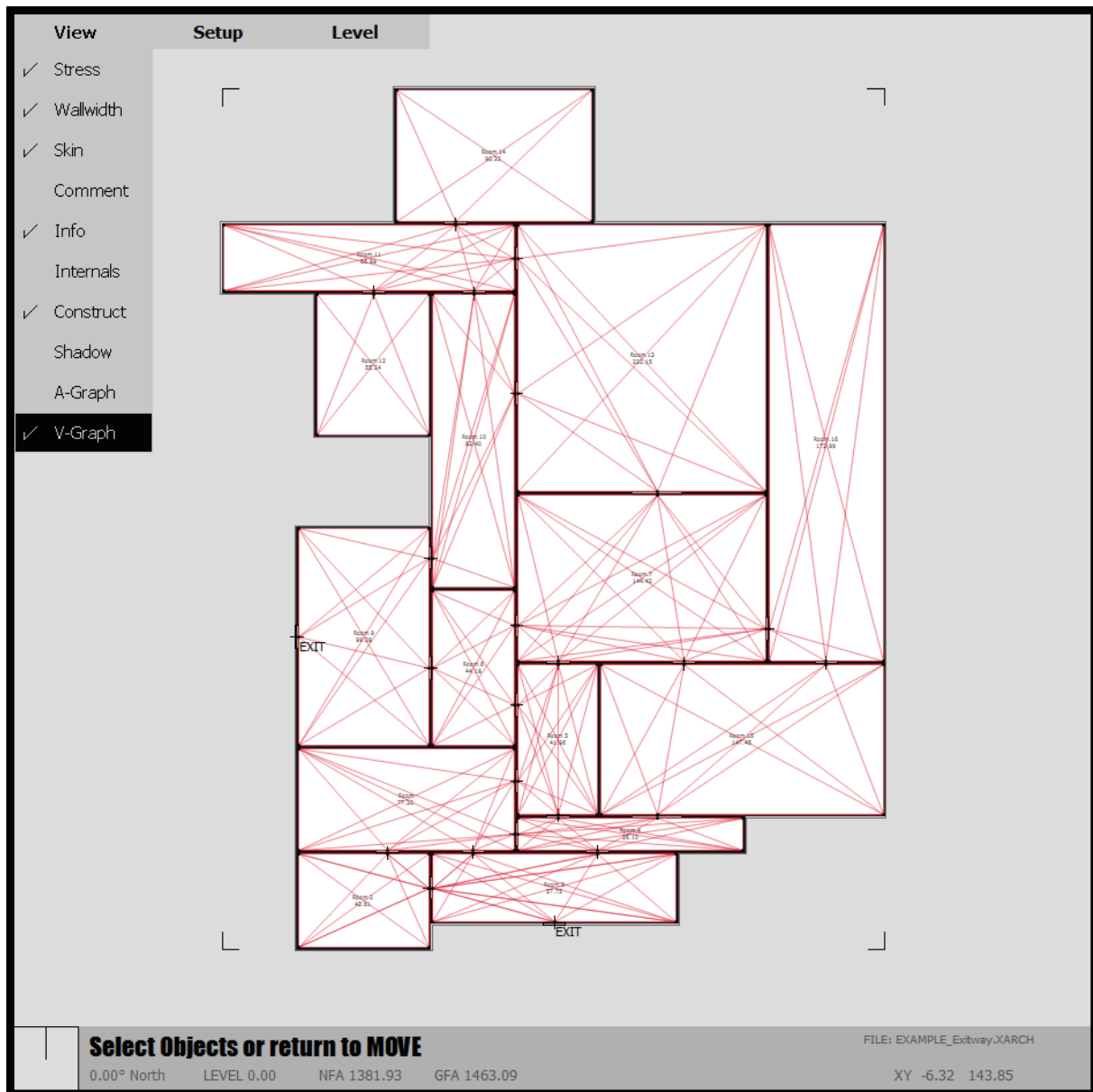


Abb 4.8 Martin Emmerer 2009, Screenshot DRAFTPAD, Visibility Graph

Aus den genannten Vorteilen wird innerhalb des vorliegenden Gebäudemodells der Sichtbarkeitsgraphen (V-Graph) als Grundlage zur Berechnung von Wegen automatisch berechnet. Die einzelnen Räume des Grundrisses bilden Teilgraphen, welche durch Brücken im Bereich der bestehenden Türen miteinander verbunden werden. Außentüren werden analog dazu als Brücken zum Aussenraum behandelt. Auch Vertikale Erschließungselemente wie Treppen oder Rampen bilden Brücken zwischen zwei Räumen. Diese werden durch die Länge der Gehlinie gewichtet (vgl. Kapitel 2.2). Innerhalb des sogenannten *Konfigurationsraums*<sup>26</sup> wird der Mensch auf einen Punkt reduziert. Dies senkt den Rechenaufwand erheblich und hat zur Folge, dass eine Rotation des Körpers bei einem Richtungswechsel keine

<sup>26</sup> In der Robotik wird ein abstrakter mehrdimensionaler Raum als Konfigurationsraum bezeichnet, dessen Dimension durch die Freiheitsgrade des beschriebenen Systems gegeben ist. Bewegungen von Menschen innerhalb eines Gebäudes haben zwei Freiheitsgrade und bewegen sich in jedem Geschöß in x- und y-Richtung. Bewegungen in z-Richtung erfolgen über vertikale Erschließungselemente wie Treppen oder Rampen. Vgl. Wikipedia, dt.: Konfigurationsraum, 25.10.2014.

Kollision mit einem Hindernis zur Folge haben kann.<sup>27</sup> Da jeder Mensch eine räumliche Ausdehnung besitzt (Abb 4.9b werden zugleich Hindernisse und Wandfluchten um 0.3m ausgedehnt, dies entspricht in etwa der Hälfte einer durchschnittlichen Schulterbreite. Dadurch kann sich der rechnerisch auf einen Punkt reduzierte Mensch den Hindernissen nur bis auf maximal 30cm nähern. Vorhandene Engstellen mit einer Durchgangslichte von weniger als 60cm können somit nicht durchlaufen werden. Abbildung 4.9a zeigt schematisch die nicht durch Menschen begehbaren Zonen eines Gebäudegrundrisses. Abbildung 4.10 zeigt relevante Teile der Graphen-Objektstruktur. Abbildung 4.11 zeigt die Objektstruktur des in DRAFTPAD verwendeten Gebäudedatenmodells.

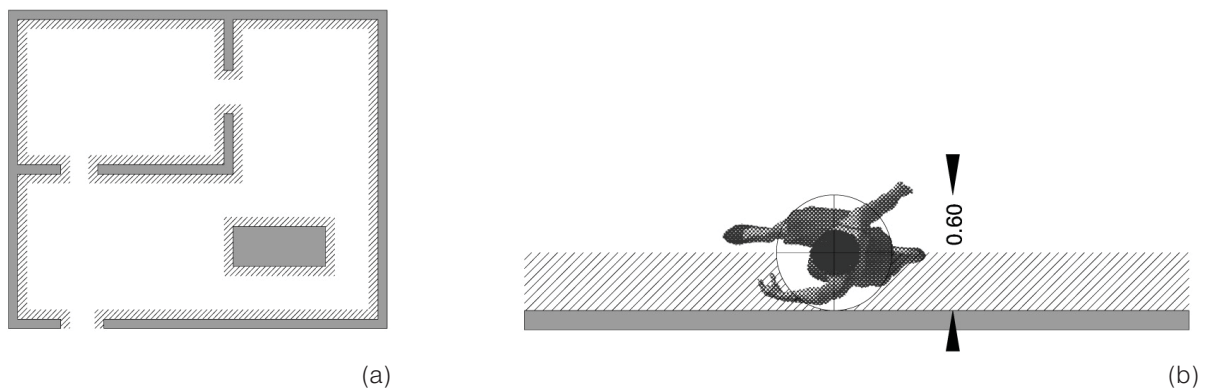


Abb 4.9 (a) Konfigurationsraum, (b) Abstraktion des menschlichen Körpers

```

TNode = record
    Center          :T2DPoint;           //Location of Node
    Parent          :integer;           //Index of Parentroom
    NType          :TGNNodeType;
    [...]
end;

TEdge = record
    Nodes :array[0..1] of integer; //Node Indices
    Index :integer;
    [...]
end;

TGraph = object
    [...]
    EList          :array of TEdge;
    NList          :array of TNode;
    NodeCount, EdgeCount :integer;
    [...]
    function GetSPath(N1,N2:integer):T3DPoly;
end;

```

Abb 4.10 Delphi Code: Datenobjekte Graph, Node, Edge

<sup>27</sup> Die Freiheitsgrade können somit auf 2 DOF (Degrees of Freedom) reduziert werden. Zur Berechnung der Kollisionsvermeidung in komplexeren Anwendungen kann die Methode der sogenannten *Minkowski-Summe* dienen. Vgl. Wikipedia, dt., Minkowski-Summe, 25.10.2014.

```
TEntities = object
  [...]
  Ro      : Array of TRoom;
  Re      : Array of TRegion;
  Wa      : Array of TWall;
  Op      : Array of TOpening;
  St      : Array of TStair;
  [...]
  V_Graph,
  Adj_Graph, Acc_Graph : TGraph;
  [...]
end;
```

Abb 4.11 Delphi Code: Objektstruktur Gebäudedatenmodell

### 4.3 Schlussfolgerungen

Das Programm DRAFTPAD zeichnet sich durch einen starken Fokus auf die frühe Entwurfsphase aus. Diese konzeptionelle Beschränkung des Detailgrades ermöglicht eine starke Vereinfachung und Reduktion der Anzahl generischer Objekte im Gebäudedatenmodell. Das Modell ist gerade deshalb auch bestens für die Untersuchungen von Gebäudeentwürfen in einem frühen Stadium geeignet. Es lässt sich beobachten, dass die bautechnische Informationsdichte in den Planungsphasen nach dem Vorentwurf zwar noch rapide zunimmt, zugleich aber die Aussagekraft eines architektonischen Gebäudeentwurfs in dieser Planungsphase in der Regel bereits ein relativ hohes Level erreicht hat. Das vorgestellte Gebäudedatenmodell verfügt über die wesentlichen Elemente, die zur korrekten Abbildung vieler, wenn auch nicht aller Gebäudeentwürfe notwendig ist. Darüber hinaus enthält es keine unnötigen Daten, welche die Untersuchungen erschweren würden.

Bei der Modellierung geometrisch besonders spezieller Gebäude wird man dennoch an die Grenzen des vorliegenden Datenmodells stoßen. Eine Modellierung eines Gebäudes ist nicht mehr sinnvoll, wenn dadurch entscheidende räumliche Qualitäten verlorengehen und die Isomorphie zu dem abgebildeten Entwurf nicht mehr gegeben ist. Ein wesentlicher Unterschied zu den üblichen Gebäudedatenmodellen besteht darin, dass die Entitäten in DRAFTPAD semantisch in Anlehnung an das architektonische Denken in der Entwurfsphase konzipiert sind. Außerdem entspricht die Objektstruktur den Elementen, durch welche Gebäude in XSAL Notation spezifiziert werden (vgl. Kapitel 3.4). Die Datenobjekte beziehungsweise Objektklassen der im Baugewerbe etablierten BIM Anwendungen bilden im Gegensatz dazu eher die zur technischen Realisierung eines Gebäudes notwendigen Bauteile ab. Durch die automatische Generierung des Access-, Adjacency- und Visibility Graph ermöglicht das vorliegende Gebäudedatenmodell auch eine direkte Anwendung bekannter Methoden zur Analyse topologischer Eigenschaften wie beispielsweise der von *Space Syntax*<sup>28</sup> (vgl. Kapitel 2.3).

---

<sup>28</sup> Vgl. Hillier/Hanson 1984.

## 5 ALGORITHMISCHE EXPLIKATION

*»Die philosophische Erkenntnis ist die Vernunftkenntnis aus Begriffen, die mathematische aus der Konstruktion der Begriffe. Einen Begriff aber konstruieren, heißt: die ihm korrespondierende Anschauung a priori darstellen.«<sup>1</sup>*

Immanuel Kant 1781

Eine Vielzahl von Entwurfparametern, wie sie in Raumprogrammen beschrieben werden, können maschinell erfasst werden. Die Spezifikationen der einzelnen Funktionsbereiche und deren Beziehungen untereinander werden bis heute üblicherweise in natürlicher Sprache verfasst. Wie schon in der Einleitung der vorliegenden Arbeit beschrieben, stellt der Wunsch nach einer „gut orientierten“ Wohnung oder die Forderung, dass Raum A „nahe“ bei Raum B situiert sein soll, für Menschen scheinbar eindeutige Planungsvorgaben dar. Wir verbinden mit dem Begriff „Orientierung“ genauso eine bestimmte Vorstellung, wie mit dem der „Nähe“. Bei genauerer Betrachtung sind selbst diese Begriffe jedoch äußerst vage. Was einen deutlichen Begriff von einem vagen unterscheidet, darauf hat schon Gottfried Wilhelm Leibniz im Jahre 1684 hingewiesen:

*»[W]ir beobachten, daß Maler und andere Künstler ganz vortrefflich erkennen, was richtig und was fehlerhaft gemacht ist, häufig aber nicht imstande sind, von ihrem Urteil Rechenschaft zu geben, und auf Befragen nur antworten, sie vermüßten in der Sache, die ihnen mißfällt, irgend etwas, sie wüßten selbst nicht was. Eine deutliche Vorstellung ist aber eine solche, wie sie die Goldscheider vom Golde haben, auf Grund von Merkmalen nämlich und Untersuchungen, die hinreichen, die Sache von allen anderen ähnlichen Körpern zu unterscheiden.«<sup>2</sup>*

Leibniz führt diesen Vorteil der Münzprüfer darauf zurück, dass ein Metall durch mehrere Merkmale eindeutig als Gold identifiziert werden kann. – Unter anderem aufgrund seines Gewichts, seiner Farbe und wegen der Eigenschaft, durch Salpetersäure nicht aufgelöst zu werden. Für den rationalistischen Denker war ein Begriff nur dann deutlich, wenn alle konstitutiven Merkmale nach vollständiger Analyse, deutlich erkannt werden konnten.

Menschen sind im intuitiven Umgang mit vagen Beschreibungen geübt. Maschinen dagegen benötigen für jede Datenverarbeitung vollständig formalisierte Informationen. Sollen architektonische Entwurfsqualitäten von Computern überwacht werden, genügt es nicht mehr eine ungefähre Vorstellung von der Bedeutung der verwendeten Begriffe zu besitzen. Dem Computer muss eine exakte Handlungsanleitung in Form eines Algorithmus bereitge-

<sup>1</sup> Kant [erstveröffentlicht 1781] 2011, 571f.

<sup>2</sup> Leibniz [erstveröffentlicht 1684] 1996, 10.

stellt werden, der beschreibt, wie jedes der durch die Begriffe bezeichneten Merkmale erhoben werden kann. Zu diesem Zweck wurde vom Autor das Konzept der *Algorithmischen Explikation* entwickelt. Hierbei handelt es sich um eine Methode, mit deren Hilfe beliebige Merkmale eines Gebäudes, von deren genauen Bedeutung anfangs nur eine vage Vorstellung besteht, systematisch, sukzessiv präzisiert werden können, bis hin zu dem Punkt, an dem die Implementierung einer Computerroutine, zur automatischen Erfassung der interessierenden Eigenschaft innerhalb eines Gebäudedatenmodells, erfolgen kann.

Im vorliegenden Kapitel wird dieser schrittweise Vorgang zuerst detailliert in allgemeiner Form dargestellt. Danach wird das Verfahren beispielhaft auf unterschiedliche Gebäudemerkmale angewandt. Die Methode der *Algorithmischen Explikation* gliedert sich im Wesentlichen in zwei Schritte, welche in den nachfolgenden Unterkapiteln 5.1 und 5.2 genauer beschrieben werden. Zunächst ist es notwendig, die häufig vage Vorstellung eines Begriffs der Alltagssprache in einen deutlichen Begriff, das heißt in eine mess- und berechenbare Angabe umzuwandeln. Danach kann der zweite Schritt erfolgen: Die Implementierung dieses Vorgangs in eine Computerroutine beziehungsweise einen Algorithmus zur Erfassung des interessierenden Merkmals in einem geeigneten digitalen Gebäudemodell. Abbildung 5.1 zeigt die Makrostruktur der Algorithmischen Explikation.



Abb 5.1 Makrostruktur der Algorithmische Explikation

## 5.1 Explikation

Die Explikation ist ein für die empirischen Wissenschaften grundlegender Vorgang. Am Beginn jeder wissenschaftlichen Arbeit sollte die exakte Bedeutung sämtlicher in die Untersuchung aufgenommenen Variablen kontrolliert werden.<sup>3</sup> In der vorliegenden Arbeit soll empirisch festgestellt werden, ob und wie weit eine architektonische Entwurfslösung das ihr

<sup>3</sup> Vgl. Bortz 2006, 60f.

zugrunde liegende Raumprogramm erfüllt. Dadurch werden alle in einem solchen Programm beschriebenen Gebäudemerkmale in diesem Sinne zu einer Variablen.

Die Explikation von Begriffen ist nicht nur in den empirischen Wissenschaften sondern auch für die Philosophie von Bedeutung. Für den deutschen Philosophen Rudolf Carnap bestand eine wichtige Aufgabe der Philosophie in der logischen Analyse der Sprache.<sup>4</sup> Carnap war Mitglied des Wiener Kreises und gilt neben Moritz Schlick, Otto Neurath und anderen als einer der Hauptvertreter des *logischen Empirismus* des 20. Jahrhunderts. Als einer der ersten Theoretiker, versuchte der Philosoph, die logischen Arbeiten von Gottlob Frege, Bertrand Russell und Alfred North Whitehead auch für erkenntnis- und wissenschaftstheoretische Fragestellungen nutzbar zu machen.<sup>5</sup>

*»Die Aufgabe der Begriffsexplikation besteht darin, einen gegebenen, mehr oder weniger unexakten Begriff durch einen exakten zu ersetzen. Der gegebene Begriff (sowie der dafür verwendete Ausdruck) soll Explikandum heißen, den exakten Begriff (sowie den dafür vorgeschlagenen Ausdruck) hingegen, der den ersten ersetzen soll, nennen wir Explikat. Das Explikandum kann der Sprache des Alltags oder einem frühen Stadium der Wissenschaftssprache entnommen sein. Das Explikat muss durch explizite Regeln für seine Anwendung gegeben werden. Dies kann z.B. durch eine Definition geschehen, welche diesen Begriff in ein bereits vorhandenes System von logischmathematischen oder empirischen Begriffen einordnet.«<sup>6</sup>*

Die Explikation von vagen Begriffen der Alltagssprache kann in zwei Teilschritten vollzogen werden. Zuerst wird ein vager Begriff, das sogenannte *Explikandum*, einer Bedeutungsanalyse unterzogen. Danach kann der Begriff durch eine operationale Definition messbar gemacht werden. Auf diese Weise gelangt man zu einem deutlichen Begriff, dem sogenannten *Explikat*.

Vager Begriff      Analytische      Operationale      Deutlicher Begriff  
Explikandum      →      Definition      →      Definition      →      Explikat

## Analytische Definition

Als erster Schritt zur Präzisierung eines Begriffs, muss dessen Bedeutung und Gebrauchsweise analysiert werden. Dies erfolgt im Zuge einer *analytischen Definition* oder *Bedeutungsanalyse*, welche von der Forscherin oder dem Forscher frei formuliert werden kann. Bei der Bedeutungsanalyse werden sämtliche sprachlichen Indikatoren eines Begriffs aufgezählt. Dabei sollten eventuell bereits vorliegende wissenschaftliche Auseinandersetzungen mit dem untersuchten Begriff berücksichtigt werden. Auch Normen oder Gesetze können in die analytische Definition eines Begriffes einbezogen werden. Die *analytische Defini-*

<sup>4</sup> Vgl. Carnap 1959, 12.

<sup>5</sup> Baumann 2006, 223.

<sup>6</sup> Carnap 1959, S.12.

tion beschreibt nicht nur, wie ein Begriff in der Sprache gebraucht wird, sondern hat auch vorschreibenden (stipulativen) Charakter. Um Begriffe für bestimmte wissenschaftliche Aufgabenstellungen verwendbar zu machen, werden nach Tadeusz Pawłowski mitunter sogar herkömmliche Bedeutungen geändert.<sup>7</sup> Auch der Psychologe und Statistiker Jürgen Bortz betont den individuellen und freien Charakter analytischer Definitionen:

*»Mit der analytischen Definition gibt der Forscher zu verstehen, was er mit einem Begriff bezeichnen will. Er legt damit gewissermaßen seine „Karten auf den Tisch“, und es bleibt nun jedermann überlassen, die analytische Definition nachzuvollziehen, oder nicht. Ob sich die Definition bewährt, bzw. ob die Definition richtig oder realistisch ist, zeigt letztlich die spätere Forschungspraxis.«<sup>8</sup>*

Wird ein Begriff zum ersten Mal, oder auf völlig neue Art definiert, spricht man von einer *Stipulativen Definition*. In der Mathematik finden sich solche Definitionen häufig. Ein bekanntes Beispiel stellt das Dreieck da. Neben dem umgangssprachlichen Begriff eines Dreiecks, existiert auch der mathematische Begriff eines Dreiecks. Letzterer wurde durch explizites Anführen der notwendigen und hinreichenden Bedingungen eingeführt, und kann daher leicht durch Angabe jener Bedingungen charakterisiert werden. Nach dem Philosophen Immanuel Kant werden solche Begriffe als *synthetisch* und deren Bildung als *Begriffskonstruktion* bezeichnet. Synthetische Begriffe, die durch stipulative Definitionen vereinbart werden, stehen in einem engen Zusammenhang mit dem, was in der Philosophie Immanuel Kants als „synthetisches Urteil a priori“ bezeichnet wird. Nach Kant handelt es sich hierbei um Urteile, die nicht auf der Basis von Erfahrung a priori gefällt werden und deren Wahrheit nicht auf der Zerlegung von Begriffen beruht, sondern in der Konstruktion neuer.<sup>9</sup>

Im Zuge einer analytischen Definition können erneut Begriffe auftauchen, deren Bedeutungen im Kontext dieser Definition wiederum zu präzisieren sind. Bis zum Vorliegen einer vollständigen Bedeutungsanalyse kann sich dieser Vorgang mehrfach rekursiv wiederholen. Danach erfolgt der nächste Schritt: Die Operationalisierung.

## Operationale Definitionen

Im Zuge der *operationalen Definition* oder *Operationalisierung* eines Begriffs werden jene Operationen aufgezählt, die zur Erfassung der, von den in der Bedeutungsanalyse aufgezählten Indikatoren abhängigen Variablen führen. Auch präzise durchgeführte analytische Definitionen eines Begriffs lassen häufig zugleich mehr als nur eine korrekte Operationalisierungen zu. Die Ausdrücke *operationale Definition* oder *Operationalisierung eines Merkmals* gehen auf den Physiker Percy Williams Bridgman zurück. Die ursprüngliche auf die Physik zugeschnittene Fassung der „Operational Analysis“<sup>10</sup> wird von Jürgen Bortz folgendermaßen zusammengefasst:

<sup>7</sup> Vgl. Pawłowski 1980, 183.

<sup>8</sup> Bortz 2006, 61.

<sup>9</sup> Vgl. Baumann 2006, 233.

<sup>10</sup> Vgl. Bridgman 1938.

1. *»Die operationale Definition ist synonym mit einem korrespondierenden Satz an Operationen (Der Begriff „Länge“ bedeutet nicht mehr und nicht weniger, als eine Reihe von Operationen, mit denen eine Länge ermittelt wird.)*
2. *Ein Begriff sollte nicht bezüglich seiner Eigenschaften, sondern bezüglich der mit ihm verbundenen Operationen definiert werden.*
3. *Die wahre Bedeutung des Begriffs findet man nicht, indem man beobachtet, was man über ihn sagt, sondern indem man registriert, was man mit ihm macht.*
4. *Unser gesamtes Wissen ist an Operationen zu relativieren, die ausgewählt wurden, um unsere wissenschaftlichen Konzepte zu messen. Existieren mehrere Sätze von Operationen, so liegen diesen auch mehrere Konzepte zugrunde.«<sup>11</sup>*

Für die Handlungen (Operationen) zur Erhebung von interessierenden Größen, welche im Rahmen empirischer Untersuchungen zu neuen Erkenntnissen führen sollen, existieren in den unterschiedlichen, wissenschaftlichen Disziplinen verschiedene Bezeichnungen. In der Physik und den Ingenieurwissenschaften spricht man von *Messung*. Dieser Vorgang ist in der Regel auf das Bestimmen einer Maßzahl als das Vielfache einer Einheit beschränkt. Im Gegensatz dazu, pflegt man laut Bortz beispielsweise in den Sozialwissenschaften mit dem Ausdruck *Messung* einen weiter gefassten Umgang. Da in diesem Bereich der Wissenschaft häufig die Maßeinheiten fehlen, begnügt man sich hier mit einer Definition der Messoperation als ein Zuordnen von Werten zu Objekten. In diesem Zusammenhang wird häufig stellvertretend für den Begriff der *Messung* von *Skalierung* gesprochen.<sup>12</sup> Die Erfassung von Datenmaterial, das in digitaler Form vorliegt, erfolgt in der Regel durch die funktionale Auswertung des Datenbestandes mit Hilfe von Algorithmen. Wenn es darum geht, Wissen in unüberschaubar großen digitalen Datenbeständen zu entdecken (Knowledge Discovery in Databases – KDD), hat sich in den letzten Jahren der Begriff des *Data-Minings* etabliert. Als Data-Mining wird die „*automatische Auswertung großer Datenmengen zur Bestimmung bestimmter Regelmäßigkeiten, Gesetzmäßigkeiten und verborgener Zusammenhänge*“<sup>13</sup> bezeichnet. Um in großen Datenbergen nach wertvollem Wissen zu suchen, werden Datenanalyse- und Entdeckungsalgorithmen eingesetzt.

Auch Bridgman verstand unter Operationen in seinen früheren Schriften ausschließlich physikalische Messvorgänge, ließ in späteren Arbeiten jedoch auch nicht-physikalische, sogenannte *Paper and Pencil* Operationen, wie Berechnungen oder geometrische Konstruktionen zu.<sup>14</sup> Die Voraussetzung dafür bestand für ihn jedoch darin, dass diese stets eindeutig ausgeführt werden konnten.<sup>15</sup> Auch Algorithmen sind exakt beschriebene Vorgehensweisen zum Lösen eines Problems in endlich vielen, eindeutig beschriebenen Schritten, und können so als Operation im Sinne Bridgmans erweiterten Operationsbegriff aufgefasst werden.<sup>16</sup>

---

<sup>11</sup> Bortz 2006, 62.

<sup>12</sup> Vgl. Bortz 2006, 65.

<sup>13</sup> Vgl. Duden online, Data-Mining, 01.07.2014.

<sup>14</sup> Vgl. Bridgman 1927; 1938, 122ff.; 1959, 522.

<sup>15</sup> Vgl. Stanford Encyclopedia of Philosophy, Operationalism, 07.08.2014.

<sup>16</sup> Vgl. Wikiwörterbuch Wiktionary, Algorithmus, 03.06.2012.



## 5.2 Implementierung

Nachdem die Bedeutung eines Gebäudemerkmals durch eine analytische Definition vereinbart und dessen Erhebungsmethode im Zuge der operationalen Definition festgelegt wurde, können die zur Erhebung der Ausprägungen des Merkmals notwendigen Operationen in eine Computerroutine (Algorithmus) implementiert werden.

*Deutlicher Begriff* → *Implementierung* → *Computerroutine*

In der vorliegenden Arbeit wird davon ausgegangen, dass die untersuchten Gebäudeentwürfe im Format des in Kapitel 4.2 spezifizierten assoziativen Gebäudedatenmodells vorliegen. Die Erfassung der interessierenden Merkmale einer Entwurfslösung erfolgt ausschließlich durch Algorithmen. Das universelle Erhebungsinstrument ist dabei in jedem Fall der Computer. Hierbei handelt es sich gewissermaßen um eine spezielle Form des Data-Minings in einem Building Information Model und soll deshalb in der Folge als BUILDING INFORMATION DATA MINING (BIDM) bezeichnet werden. Die Implementierung in eine Computerroutine erfolgt durch die Programmierung eines „Entdeckungsalgorithmus“ in Form der Programmfunktion *Discover*.

### Entdeckungsalgorithmen

In der Informatik werden Programmkonstrukte, die wiederholt verwendbare Funktionalitäten eines Computerprogramms enthalten, als *Funktionen* bezeichnet. Im Gegensatz zu Prozeduren, liefert eine Funktion direkt einen Rückgabewert, das sogenannte Resultat. Gemäß dem Funktionskonzept in der Mathematik, stellt auch eine Programmfunktion eine Abbildungsvorschrift dar, welche durch eine Reihe von operationalen Definitionen beschrieben wird. Die zur Berechnung des Resultats benötigten Daten werden in Form von *Argumenten* ( $A_1 - A_n$ ) übergeben.

Soll beispielsweise die Ausprägung eines bestimmten Merkmals eines Raums in einem digitalen Gebäudemodell erhoben werden, wird eine Funktion gebildet, der die betreffende Instanz jenes Datenobjekts als Argument (in Form eines Pointers) übergeben wird, welches den entsprechenden Raum informationstechnisch repräsentiert. Soll ein Beziehungsmerkmal zwischen zwei Räumen festgestellt werden, so müssen der Funktion beide involvierten Räume als Argumente übergeben werden (vgl. Einleitung Prädikatenlogik). Funktionen können im Prinzip wie Variablen verwendet werden.

$$\textit{Result} := \textit{Discover}(\textit{Model}, R_1) \quad \textit{Result} := \textit{Discover}(\textit{Model}, R_1, R_2)$$

Bei der Erhebung von Merkmalsausprägungen bestimmter Räume in einem digitalen Gebäudemodell genügt es häufig nicht, nur die entsprechenden Räume als Argumente zu übergeben. Soll zum Beispiel die Entfernung zweier Räume festgestellt werden, so müssen auch alle zwischen Start- und Zielraum liegenden Räume in die Berechnung miteinbezogen werden, da eine vollständige Route durch das Gebäude berechnet werden muss. Darüber

hinaus, ist in einem solchen Fall auch ein Zugriff auf andere Objekte im Datenmodell, wie Türen oder Treppen, notwendig. Zur Erhebung vieler Merkmale muss der Funktion das gesamte Datenmodell global zur Verfügung gestellt beziehungsweise als Argument übergeben werden.

Spätestens zum Zeitpunkt der Implementierung ist zu entscheiden, auf welchem Skalenniveau das betreffende Merkmal erhoben werden soll. Das bedeutet nichts anderes, als dass die Menge der möglichen Merkmalsausprägungen bestimmt werden muss. Die Funktion liefert in der Folge für jedes als Argument übergebene Raum-Datenobjekt genau ein Element der Menge der möglichen Merkmalsausprägungen als Resultat, zurück.

Bei der Erhebung von Merkmalsausprägungen auf der Verhältnisskala ist das Resultat der Funktion eine reelle Zahl (Real), auf Absolutskalenniveau eine natürliche Zahl (Integer). Zudem ist bei der Erfassung quantitativer Merkmale das Festsetzen einer Einheit notwendig. Bei Messfunktionen für qualitative Merkmale, das heißt auf Ordinal- oder Nominalskalenniveau sind die Rückgabewerte Kategorien, welche durch Begriffe (Strings) bezeichnet werden. Für die Datenverarbeitung ist es vorteilhaft, jeder Kategorie wie beispielsweise „gut“, „mittel“ oder „schlecht“ eine natürliche Zahl (0, 1, 2) als Index zuzuweisen. Durch eine solche Kodierung der Wertemenge erhalten die Kategorien numerische Stellvertreter, die beim Aufruf der Funktion als Resultat zurückgegeben werden. Abbildung 5.2 zeigt beispielhaft Funktionen, die Merkmale auf vier unterschiedlichen Skalenniveaus erheben.

```
function Area(Room):Real                ...Verhältnisskala

function WindowCount(Room):Integer      ...Absolutskala

function Orientation(Room):Integer      ...Ordinalskala
    0...good
    1...middle
    2...bad

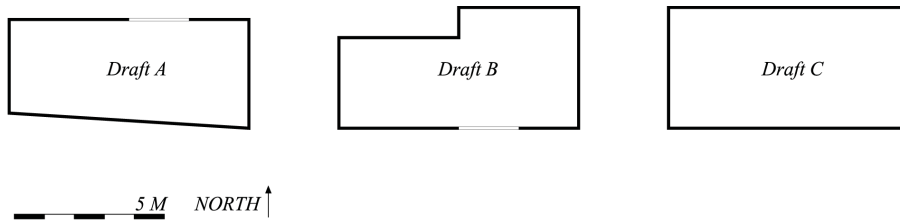
function Figure(Room):Integer           ...Nominalskala
    0...Triangle
    1...Rectangle
    2...L-Shape
    3...T-Shape
    4...Cross
    5...Star
    6...Other
```

Abb 5.2 Funktionen für 4 Skalenniveaus

## Beispiel

Die maschinelle Erhebung von Merkmalsausprägungen soll nun anhand eines einfachen Beispiels erläutert werden. In drei Entwurfsvarianten (Abb 5.3a) eines einzelnen Raums werden mit Hilfe der in Abbildung 5.2 dargestellten Programmfunktionen - *Area*, *Windowcount*, *Orientation* und *Figure* - die konkreten Ausprägungen der interessierenden Merkmale maschinell erhoben. Die Resultate der Entdeckungsalgorithmen bilden für jeden Raum einen Merkmalsvektor mit vier Dimensionen (Abb 5.3b).

(a)



$$v_A = \begin{Bmatrix} 28.0 \\ 1 \\ 2 \\ 6 \end{Bmatrix} \quad v_B = \begin{Bmatrix} 29.0 \\ 1 \\ 0 \\ 2 \end{Bmatrix} \quad v_C = \begin{Bmatrix} 31.0 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \quad \begin{array}{l} \textit{Area} \\ \textit{Windowcount} \\ \textit{Orientation} \\ \textit{Figure} \end{array} \quad (b)$$

Abb 5.3 (a) Drei Entwurflösungen (b) Merkmalsvektoren

## Subroutinen

Das Funktionskonzept in der Programmierung ermöglicht es auch, dass eine Funktion innerhalb einer anderen Funktion ausgeführt wird. Dadurch ist es möglich, basale ARCHITEKTUR ROUTINE(n) innerhalb komplexerer Routinen aufzurufen. Ein einfaches Beispiel hierfür ist der *Fensterquotient*, der das Verhältnis von Fensterflächen zur Nettonutzfläche eines Raums prozentuell wiedergibt. Hat man also bereits die ARCHITEKTUR ROUTINE(n) *Net Area* und *WindowArea* gebildet, können diese von der Funktion *WindowQotient* zur Berechnung des Resultats benutzt werden. Auf diese Weise lassen sich komplexere Begriffe aus basalen Begriffen bilden und Redundanzen vermeiden.

```
function WindowQuotient(Room):Real
begin
  Result:=WindowArea(Room)/NetArea(Room);
end;
```

Abb 5.4 Programmfunktion *Window Quotient*

Der Einsatz von Subroutinen bietet auch die Möglichkeit eines *Downgrades* von Merkmalen, indem hohe Skalenniveaus auf ein niedrigeres Niveau gebracht werden. Wie schon in Kapitel 3.2 erklärt wurde, ist mit dem höchsten Skalenniveau nicht immer auch die höchste Aussagekraft verbunden. Ein Beispiel hierfür wäre eine ARCHITEKTUR ROUTINE *OrderOfMagnitude* welche beliebige Raumgrößen in nur drei Kategorien „small“ (kleiner 60m<sup>2</sup>), „medium“ (60 bis 80 m<sup>2</sup>) und „large“ (größer als 80 m<sup>2</sup>) ordinalskaliert erfasst. Diese Routine kann

wiederum auf die verhältnisskalierte Funktion *NetArea* zugreifen, die Kategorien (small, medium, large) anwenden und ein Ergebniskategorie als ganzzahliges Ergebnis (Index) zurückliefern.

```
function OrderOfMagnitude (Room):Integer

var A:Real;

begin
  A:=NetArea(Rom);
  if A<60 then Result:=0
    else if A>80 then Result:=2
      else Result:=1;
end;

0...small
1...medium
2...large
```

Abb 5.5 Programmfunktion *Order of Magnitude*

## Zusammenfassung

In der vorliegenden Arbeit wird die vollständig dokumentierte ALGORITHMISCHE EXPLIKATION bestehend aus Explikation und Implementierung eines interessierenden Gebäudemerkmals als ARCHITEKTUR ROUTINE bezeichnet.

Um eine klare Strukturierung und Sicherung der Vollständigkeit jeder ARCHITEKTUR ROUTINE zu gewährleisten, wird die *Begriffskonstruktion* immer im selben charakteristischen Layout verfasst. Das in Abbildung 5.6 dargestellte Formular enthält alle nötigen Datenfelder zur Spezifikation des betrachteten Gebäudemerkmals sowie freie Bereiche für die Dokumentation in Text- und Skizzenform und für den Sourcecode der Implementierung. Da jede ARCHITEKTUR ROUTINE den Charakter eines eigenständigen, wissenschaftlichen Papers hat, werden Literaturverweise jeweils gesondert als Referenzen angegeben. Durch die wiederholte Anwendung des Verfahrens der ALGORITHMISCHEN EXPLIKATION auf möglichst viele Begriffe, welche alltäglich zur Beschreibung von Entwurfsaufgaben herangezogen werden, entsteht ein Vokabular, welches nicht nur an intersubjektive Vorstellungen geknüpft ist, sondern auch einen Dialog mit der Maschine erlaubt. Abbildung 5.8 zeigt das Inhaltsverzeichnis des im Rahmen dieser Arbeit begonnenen Merkmalskatalogs. Die dazugehörigen, vollständigen Begriffsexplikationen und deren Implementierung sind im Anhang ab Seite 135 zu finden.

## Property



№ -  
Entity



Autor: -  
Datum: -  
Skala: -  
Einheit: -  
Wertemenge: -

Vokabular:  
(0) -  
(1) -  
(n) -

### (a) Analytische Definition

...

### (b) Operationale Definition

...

### (c) Implementierung

...

Abb 5.6 Formular-Layout Architektur Routine

## 5.3 Schlussfolgerungen

Mit dem Durchlaufen der ALGORITHMISCHEN EXPLIKATION erfährt *das geheimnisvolle Band der Assoziation*<sup>17</sup>, welches natürlichsprachige Begriffe mit den Objekten und Phänomenen der realen Welt verbindet, ein maschinelles Pendant. Bei operationalisierten Begriffen (Explikaten) rücken präzise festgelegte Operationen an die Stelle des subjektiven, beim Spracherwerb und in der Kommunikation mit anderen Personen erworbenen, gewohnheitsmäßigen Umgang mit vagen Begriffen (Explikandum). Dabei wird stets nach dem gleichen Schema vorgegangen. Die Bildung jeder ARCHITEKTUR ROUTINE erfolgt immer gemäß den, in Abbildung 5.7 dargestellten Teilschritten, welche die sukzessive Umwandlung eines anfangs vagen Begriffs der Alltagssprache zuerst in eine mess- und berechenbare Angabe und schlußendlich in einen Algorithmus vollziehen.

<sup>17</sup> Vgl. Saussure 2001, 77.

Es wird nicht immer gelingen, Lösungen mit gleich hoher objektiver oder zumindest intersubjektiver Gültigkeit (Invariancy) zu finden. Auch die im Anhang dieser Arbeit (S.135) vorgestellten Routinen haben nicht alle denselben Gültigkeitsgrad. Oft kann der Grad der Gewissheit, Gültigkeit oder Wissenschaftlichkeit nur geschätzt werden. Dazu wird vom Autor ein ähnliches 5-Sterne System eingeführt, wie es im Grunde schon Christopher Alexander bei den Entwurfsmustern seiner *Pattern Language* einsetzte.<sup>18</sup> Durch die Kennzeichnung mit fünf Sternen drückt die Autorin oder der Autor einer ARCHITEKTUR ROUTINE aus, zu glauben, eine echte Invariante gefunden zu haben, was bedeutet, dass die Möglichkeit einer anderen richtigen Erfassung des beschriebenen Merkmals ausgeschlossen wird. Eine Bewertung mit drei Sternen deutet an, dass dem Ziel der Identifikation einer Invariante zumindest nahe gekommen wurde, jedoch nicht ausgeschlossen wird, dass sich die vorgestellte Lösung durch ein noch sorgfältigeres Studium des Problems weiter verbessern lässt. Nur ein Stern signalisiert hingegen, dass sich die Autorin oder der Autor einer ARCHITEKTUR ROUTINE ziemlich sicher ist, dass es andere, „richtigere“ oder „bessere“ Lösung für die Operationalisierung des betreffenden Merkmals geben müsste. Zwei oder vier Sterne drücken Nuancen zwischen diesen Zuständen aus.

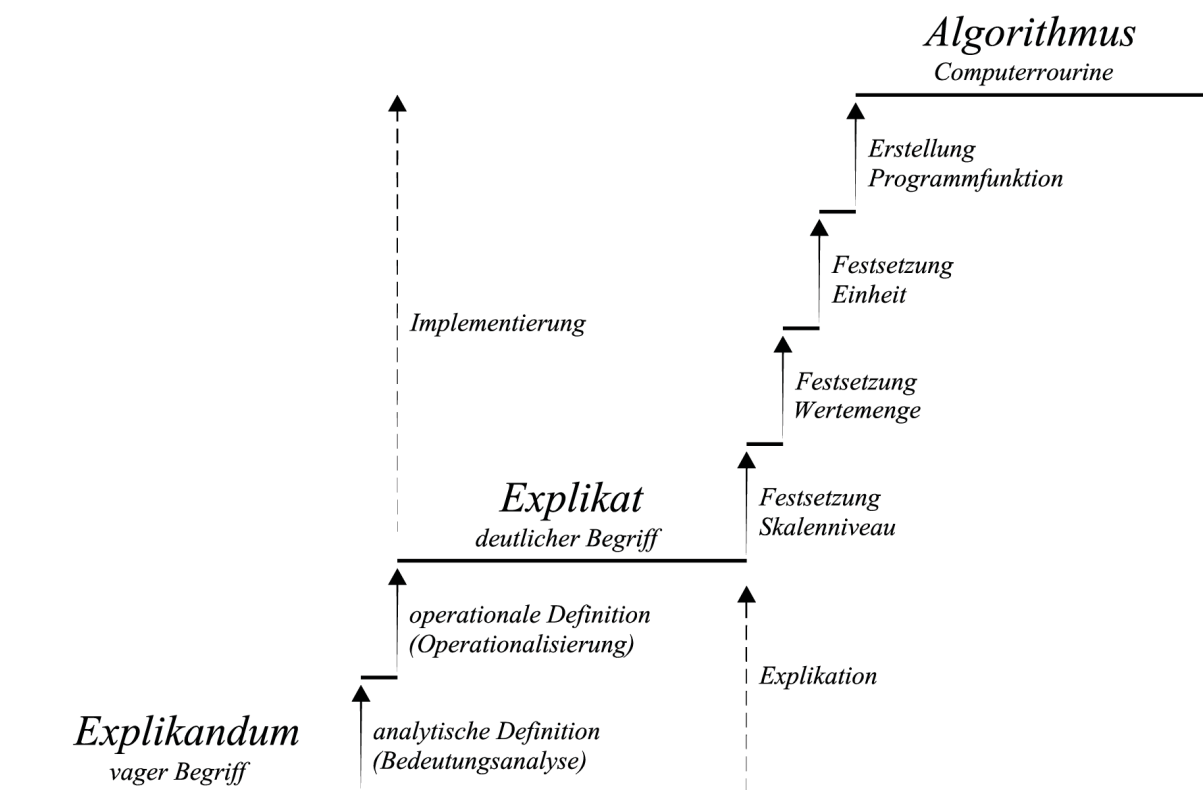


Abb 5.7 Übersicht Algorithmische Explikation

<sup>18</sup> Vgl. Alexander 1995, XIV ff.

## 5.4 Inhaltsverzeichnis Merkmalskatalog\*

No	Caption	Entity	Scale	Type
1	<i>Net Area (m)</i>	Room	V	Nettofläche
2	<i>Group Net Area</i>	Group (Rooms)	V	Gruppenfläche netto
3	<i>Order of Magnitude</i>	Room	O	Größenordnung
4	<i>Orthogonality</i>	Room	N	Orthogonal oder nicht
5	<i>Floor Plan Figure</i>	Room	N	Grundrissfigur
6	<i>Overview</i>	Room	V	Einsichtigkeit, Überschaubarkeit
7	<i>Window Count</i>	Room	A	Anzahl der Fenster
8	<i>Door Count</i>	Room	A	Anzahl der Zugänge
9	<i>Window Area</i>	Room	A	Gesamte Fensterfläche
10	<i>Window Quotient</i>	Room	V	Gesamte Fensterfläche/ Nettofl.
11	<i>Quality of Orientation</i>	Room	O	Qualität der Orientierung (Licht)
12	<i>Morning Sun</i>	Room	N	Morgensonne, Ostorientierung
13	<i>Light from two Sides</i>	Room	N	Licht von mehr als einer Seite Pattern Language No159
14	<i>Escape Way Length (m)</i>	Room	V	Fluchtwegslänge
15	<i>Integration</i>	Room	V	Space Syntax
16	<i>Walk Distance (m)</i>	Room - Room	V	kürzeste fußläufige Entfernung
17	<i>Direct Access (Doorway)</i>	Room - Room	N	Zugänglichkeit (Türverbindung)
18	<i>Adjacency</i>	Room - Room	N	Nachbarschaft
19	<i>Same Level</i>	Room - Room	N	im selben Geschoß
.				
.				
.				
n				

V ... Verhältnisskala

A ... Absolutskala

N ... Nomialskala

O ... Ordinalskala

\* Vollständiger Katalog siehe Anhang S.135

## 6 ANWENDUNGEN

Gebäudemerkmale, die durch das in Kapitel 5 vorgestellte Verfahren der ALGORITHMISCHEN EXPLIKATION operationalisiert wurden, können als Instanzen einer abstrakten Klasse im Sinne der objektorientierten Programmierung modelliert werden. Mit Hilfe der in Abbildung 6.1 schematisch dargestellten Objektklasse *ArchRoutine* wird ein interessantes Gebäudemerkmal durch die für die weitere Datenverarbeitung relevanten Attribute beschrieben. Zusätzlich verfügt die Klasse über zwei aktive Programmfunktionen: Die erste, *Discover* - der Entdeckungsalgorithmus welcher beschreibt, wie das Merkmal im Gebäudedatenmodell erhoben wird, ist in Kapitel 5 bereits ausführlich beschrieben worden. Die zweite Funktion mit der Bezeichnung *Satisfaction* dient dazu, den Zielerfüllungsgrad der durch die Funktion *Discover* erhobenen Größen, entsprechend einer Präferenzfunktion zu bestimmen.

<i>Class Name</i> →	<b><i>ArchRoutine</i></b>
<i>Properties</i> →	<p><b><i>Denotation:</i></b>    <i>String</i></p> <p><b><i>Definition:</i></b>    <i>String</i></p> <p><b><i>Scaletype:</i></b>    <i>Scale of Measurement</i></p> <p><b><i>Vocabulary:</i></b>    <i>String Array</i></p> <p><b><i>Unit:</i></b>            <i>String</i></p> <hr/> <p><b><i>Model:</i></b>            <i>Pointer</i></p>
<i>Methods</i> →	<p><b><i>Discover():</i></b>      <i>Real or Integer</i></p> <p><b><i>Satisfaction():</i></b> <i>Percent {0..1}</i></p>

Abb 6.1 abstrakte Klasse

In den folgenden Unterkapiteln wird der Prozess der maschinellen Evaluierung von Entwurfslösungen durch den Einsatz der ARCHITEKTUR ROUTINE(n) vorgestellt. In Kapitel 6.1 wird die rechnerische Zusammenfassung (Aggregation) aller Teilziele einer Gebäudespezifikation zu einer integralen Fitnessfunktion (Objective Function) beschrieben. Die Aggregation erfolgt auf zwei Hierarchieebenen: Zuerst müssen die einzelnen Zielerfüllungsgrade (Satisfaction) für jede, der in der XSAL-notierten Gebäudespezifikation beschriebenen Eigenschaften der Objekte im Gebäudedatenmodell, wie Raumgruppen, Räume, Zonen und Beziehungen berechnet werden. Aus diesen kann der *Fitnesswert* jedes Objekts ermittelt werden. Danach können diese Werte in einem zweiten Schritt zu einem *Gesamtfitnessfaktor* zusammengefasst werden. Die allgemeine Beschreibung dieses Evaluationsprozesses wird zur Veranschaulichung durch ein einfaches Beispiel begleitet. Zur Visualisierung der teilweise komplexen und wenig anschaulichen mathematischen Berechnungen, werden ge-



eignete grafische Darstellungsmittel herangezogen werden. Schlussendlich wird in Kapitel 6.2 anhand der vom Autor entwickelten Software ARCHILL.ES gezeigt, wie unter diesen neuen Voraussetzungen, durch die Einbindung von ARCHITEKTUR ROUTINE(n) eine Entwurfsumgebung als Schnittstelle zwischen Mensch und Maschine konkret aussehen kann. Die Funktionalitäten, dieser für den architektonischen Entwurf maßgeschneiderten Softwareumgebung, werden durch eine Reihe praktischer Anwendungsbeispiele illustriert.

## 6.1 Fitness einer Entwurfslösung

Im Grunde geht es in der Folge darum, Entwurfslösungen oder Teilen davon Zahlenwerte zwischen 0.0 und 1.0 zuzuordnen, welche den Grad der Erfüllung der in einer Gebäudespezifikation beschriebenen Zielvorstellungen ausdrücken. Wenn diese Grundvoraussetzung geschaffen werden kann, ist es auch möglich, in einer Menge von Entwurfsvarianten automatisch die passendste Alternative zu identifizieren. In der Entscheidungstheorie spricht man in solchen Fällen vom *Nutzwert* einer Alternative. Im Zusammenhang mit *evolutionären Algorithmen* (Evolutionary Computing) hat sich der aus der Populationsgenetik stammende Begriff der *Fitness* etabliert. Der ursprüngliche biologische Fitnessbegriff misst die Angepasstheit eines Individuums an seine Umwelt.<sup>1</sup> Übertragen auf die maschinelle Evaluierung von Entwurfsvarianten gilt: Je eher eine Entwurfsvariante die in der Gebäudespezifikation formulierten Ziele erreicht, desto höher ist ihre Fitness. Aus den im Zentrum dieser Arbeit stehenden ARCHITEKTUR ROUTINE(n) kann für ein Gebäude die Fitnessfunktion, das Kernstück jedes evolutionären Algorithmus, modular zusammengesetzt werden. Unabhängig von den Mutations- und Rekombinations-Operationen, also dem eigentlichen evolutionären Algorithmus selbst, welcher kein Thema der vorliegenden Arbeit darstellt, muss in der Fitnessfunktion das gesamte für die Bewertung der Qualität einer Entwurfsvariante benötigte Wissen, gekapselt werden. Das in der Folge vorgestellte Verfahren zur Ermittlung der Fitness einer Entwurfslösung, stützt sich im Wesentlichen auf die Methode der *Nutzwertanalyse* aus dem Jahre 1970 von Christof Zangemeister, Professor am Institut für Psychologie und Arbeitswissenschaft der TU Berlin.

*»Nutzwertanalyse ist die Analyse einer Menge komplexer Handlungsalternativen mit dem Zweck, die Elemente dieser Menge entsprechend den Präferenzen des Entscheidungsträgers bezüglich eines multidimensionalen Zielsystems zu ordnen.«<sup>2</sup>*

Die Nutzwertanalyse ist ein transparentes und nachvollziehbares Mittel zur Unterstützung von Entscheidungsprozessen, in denen neben qualitativen, also durch Zahlen erfassbaren Kriterien, auch *weiche* Faktoren oder Qualitäten zu berücksichtigen sind. Diese als *Scoring Modell* bekannte Methode hat mittlerweile als Bewertungsverfahren zur Unterstützung von Entscheidungen auch in vielen Gebieten der Planung (Infrastrukturplanung, Regionalplanung<sup>3</sup> etc.) weite Verbreitung gefunden.<sup>4</sup> So wird sie beispielsweise bei der Entscheidung für die Wahl neuer Standorte von Unternehmen eingesetzt.

---

<sup>1</sup> Vgl. Barker 2009.

<sup>2</sup> Zangemeister 1970, 45.

<sup>3</sup> Vgl. Strassert 1995.

Nutzwertanalysen können in Tabellenkalkulationsprogrammen erstellt werden. Es existieren aber auch eigene Decision Support Softwareapplikationen, welche spezielle Module zur Durchführung von Nutzwertanalysen enthalten. Beispiele hierfür sind *CelsiEva*<sup>5</sup> oder der *Decision Advisor*<sup>6</sup>. Zur Durchführung einer Nutzwertanalyse müssen zumindest zwei Bedingungen erfüllt sein: Erstens wird davon ausgegangen, dass der Erfüllungsgrad jedes Teilziels (Zielerfüllungsgrad) quantitativ ermittelt werden kann. Zweitens müssen die einzelnen Zielerfüllungsgrade so aufeinander abgestimmt sein, dass bei gleicher Gewichtung der Merkmale, eine mittlere Bewertung bezüglich eines Kriteriums keinen höheren Erfüllungsgrad aufweist, als eine gute Bewertung eines anderen.<sup>7</sup> Diese beiden Grundvoraussetzungen zur Anwendung der Nutzwertanalyse bei der Evaluierung von Entwurfslösungen sind durch die ALGORITHMISCHE EXPLIKATION und den Einsatz von Präferenzfunktionen erfüllt.

Nach der maschinellen Erhebung der Ausprägungen interessierender Merkmale von Datenobjekten wie Raumgruppen, Räumen, Zonen oder räumlichen Beziehungen durch die entsprechenden ARCHITEKTUR ROUTINE(n) kann die Berechnung des Zielerfüllungsgrades mit Hilfe der Präferenzfunktionen erfolgen. Bei diesem Vorgang werden alle Merkmalsausprägungen über die Normierungsfunktion *Satisfaction* auf das Intervall [0..1] abgebildet.

## Zielerfüllungsgrad eines Gebäudemerkmals

Der Vorgang zur Ermittlung des sogenannten Zielerfüllungsgrades (*Satisfaction*) eines einzelnen XSAL-notierten Entwurfsmerkmals (Teilziel) in einem Gebäudedatenmodell lässt sich durch eine einfache geometrische Konstruktion darstellen: Die in der Gebäudespezifikation definierte Präferenzfunktion wird an der Stelle  $x$ , welche im Datenmodell durch eine ARCHITEKTUR ROUTINE erhobene Ausprägung eines interessierenden Merkmals repräsentiert, mit einer vertikalen Geraden geschnitten. Dadurch erhält man den Schnittpunkt XP, dessen  $y$ -Koordinate dem Zielerfüllungsgrad  $f(x)$  entspricht. Gemäß den Definitionen der XSAL-Notation sind die  $y$ -Koordinaten von Präferenzfunktionen auf ein Intervall reeller Werte von 0 bis 1 beschränkt. Im Grunde handelt es sich hierbei um einen Normierungsvorgang bei dem beliebige Skalen auf ein Intervall von 0.0 bis 1.0 abgebildet werden. Abbildung 6.2 zeigt diesen Vorgang schematisch anhand einer symbolischen Präferenzkurve. Die in XSAL definierten Präferenzfunktionen haben immer einen Start- und Endpunkt. Da die Merkmalsausprägungen quantitativer Merkmale nicht auf ein Intervall beschränkt sind, kann es vorkommen, dass der Schnittpunkt XP außerhalb des, durch Punkte definierten Polygonzugs liegt. Entsprechend den Definitionen des in Kapitel 3.4 vorgestellten XSAL-Standards wird angenommen, dass Start- und Endpunkt PS und PE für absolut- und verhältnisskalierte Merkmale eine theoretische Fortsetzung in Form eines horizontalen Strahls finden. Abbildung 6.3 zeigt den Pseudocode der Programmfunktion *Satisfaction* zur Ermittlung des Zielerfüllungsgrades.

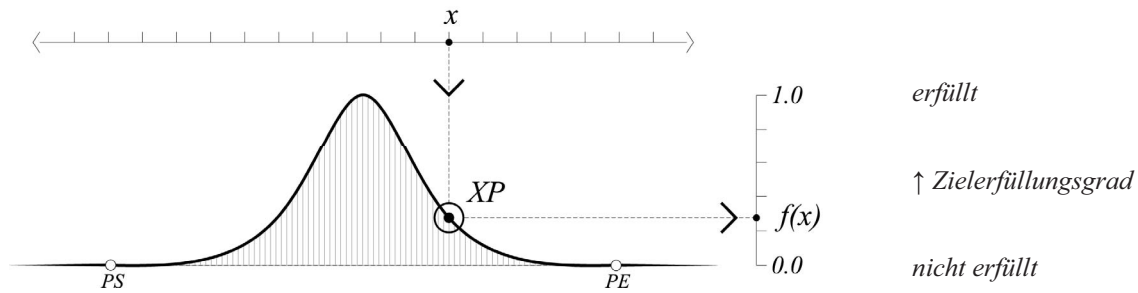
---

<sup>4</sup> Müller-Herbers 2007, 42.

<sup>5</sup> Celsi Ag: [www.celsi.ch](http://www.celsi.ch)

<sup>6</sup> PBroker AG: [www.pbroker.ch](http://www.pbroker.ch)

<sup>7</sup> Vgl. Gerdes u.a. 2004, 73.



→ Merkmalsausprägung  $x$

Abb 6.2 Präferenzfunktion: Ermittlung des Zielerfüllungsgrades

```
function Satisfaction(Valuefunction, X):Real
begin
  SP = starting point of Valuefunction
  EP = end point of Valuefunction
  X1 = X-Coordinate of SP
  Y1 = Y-Coordinate of SP
  X2 = X-Coordinate of EP
  Y2 = Y-Coordinate of EP
  if (X<X1) then Result = Y1
  if (X>X2) then Result = Y2
  if (X1<X) and (X<X2) then
    XP = Intersection(Valuefunction, X)
    Result = Y-Coordinate of XP
end;
```

Abb 6.3 Pseudocode Funktion *Satisfaction*

Bei der Evaluierung der Datenobjekte des in Kapitel 4.2 spezifizierten Gebäudedatenmodells, welche Raumgruppen, Räume, Raumzonen oder räumliche Beziehungen einer Entwurfslösung repräsentieren, werden in der Regel für ein Objekt zugleich mehrere Merkmale erhoben. Dies soll in der Folge anhand des Beispiels eines einzelnen Raums dargestellt werden, welcher in XSAL-Notation durch das quantitative Merkmal *Net Area*, sowie die qualitativen Merkmale *Quality of Orientation* und *Figure* spezifiziert wird. Anhand der explizit beschriebenen Merkmale und Präferenzfunktionen ist zu erkennen, dass es sich bei der in Abbildung 6.4 dargestellten XSAL-notierten Gebäudespezifikation um den Wunsch nach einem ca. 30m<sup>2</sup> großen Raum handelt, der im Grundriss die Form eines Rechtecks aufweisen sollte und sich möglichst nach Süden oder Westen orientiert.

```
<room name="Room">
  <attribute>
    <caption>Quality of Orientation</caption>
    <description>...</description>
    <scaletype>Ordinal</scaletype>
    <vocabulary>High;Medium;Low;None</vocabulary>
    <valuefunction>
      <vertex>0,1.00</vertex>
      <vertex>1,0.67</vertex>
      <vertex>2,0.33</vertex>
      <vertex>2,0.00</vertex>
    </valuefunction>
  </attribute>
</attribute>
```

```

<caption>Area Net</caption>
<description>...</description>
<scaletype>Ratio</scaletype>
<unit>m2</unit>
<valuefunction>
  <vertex>15.0,0.0</vertex> // -50%
  <vertex>30.0,1.0</vertex>
  <vertex>45.0,0.0</vertex> // +50%
</valuefunction>
</attribute>
<attribute>
  <caption>Figure</caption>
  <description>...</description>
  <scaletype>Nominal</scaletype>
  <vocabulary>Triangle;Rectangle;L-Shape;Other</vocabulary>
  <valuefunction>
    <vertex>0,0.0</vertex>
    <vertex>1,1.0</vertex>
    <vertex>2,0.0</vertex>
    <vertex>3,0.0</vertex>
  </valuefunction>
</attribute>
</room>

```

Abb 6.4 XSAL Spezifikation

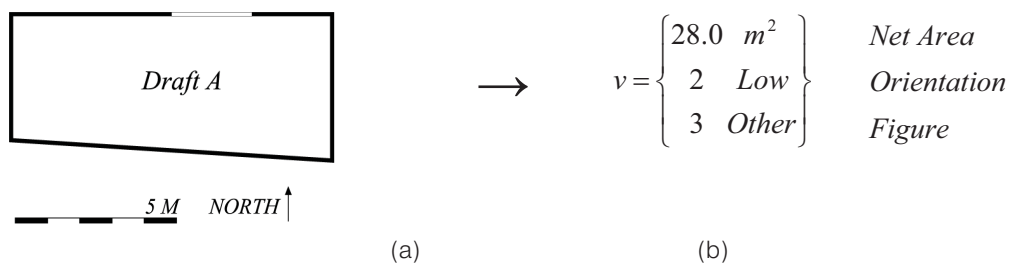


Abb 6.5 (a) Entwurfslösung (b) Merkmalsvektor

In Abbildung 6.5a ist der Grundriss einer fiktiven Entwurfslösung schematisch dargestellt. Es handelt sich um einen Raum mit einem Fenster im Norden, dessen polygonale Figur zwei rechte Winkel und eine leicht schräggestellte Kante aufweist. Die Ausprägungen der drei interessierenden Merkmale *Net Area*, *Orientation* und *Figure* wurden mit Hilfe der jeweiligen ARCHITEKTUR ROUTINE(n) erfasst (s. Anhang S.135). Die Messergebnisse bestehen entsprechend den Skalenniveaus der Merkmale aus reellen oder natürlichen Zahlen: Net Area: 28.0m<sup>2</sup>, Quality of Orientation: Low=2, Figure: Other=3. Die interessierenden Eigenschaften des Raums A bilden zusammen den *Merkmalsvektor*  $v$  (Abb 6.5b). Jedes Merkmal wird durch eine Dimension dieses Vektors repräsentiert. Anstatt des Raumobjekts selbst, wird bei der maschinellen Evaluierung von Entwurfslösungen nur ein Vektor, bestehend aus einer endlichen Zahl numerisch kodierter Merkmale betrachtet, welcher die Serie der interessierenden, durch ARCHITEKTUR ROUTINEN operationalisierten Eigenschaften zusammenfasst und so einen sogenannten *Merkmalsraum*<sup>8</sup> aufspannt.

<sup>8</sup> Ein Merkmalsraum ist ein mathematischer Raum, der ein Objekt durch dessen Messwerte in Bezug auf dessen besondere Eigenschaften bzw. Merkmale bestimmt. Vgl. Wikipedia dt., Merkmalsraum, 10.09.2014.

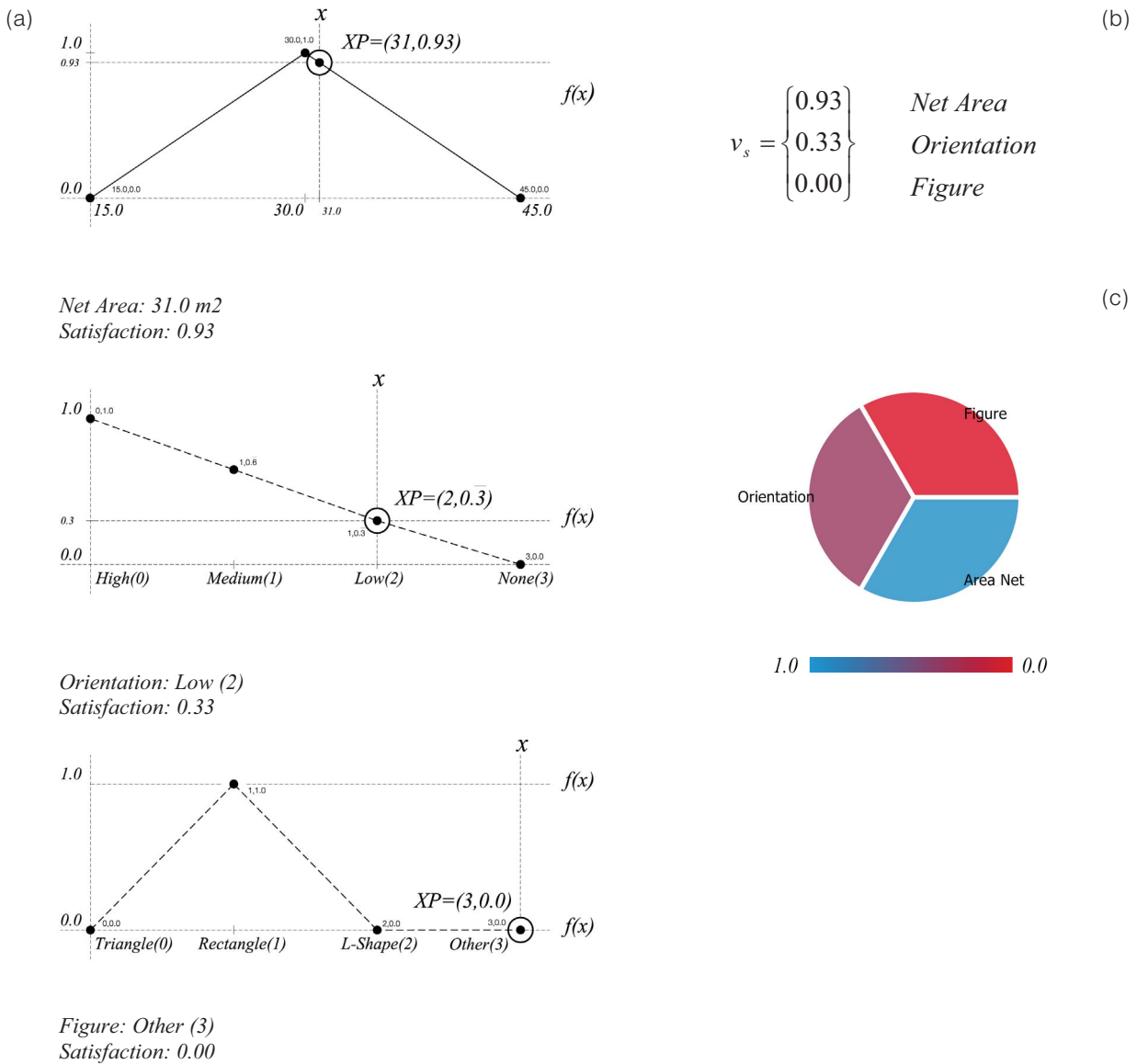


Abb 6.6 (a) Konstruktion Satisfaction (b) Satisfaktionsvektor (c) Eigenschaftsprofil

Mit Hilfe der XSAL-notierten Präferenzfunktionen, kann der Zielerfüllungsgrad (Satisfaction) jedes in der Gebäudespezifikation beschriebenen Raummerkmals berechnet werden. Abbildung 6.6a zeigt den Konstruktionsvorgang für die drei Merkmale *Net Area*, *Orientation* und *Figure*, welcher durch die Funktion *Satisfaction* (Abb 6.3) der jeweiligen ARCHITEKTUR ROUTINE automatisch ausgeführt wird. Die daraus resultierenden Werte 0.93, 0.33 und 0.00 bilden einen, auf das Intervall [0..1] normierten Vektor  $v_s$ , welcher als *Satisfaktionsvektor* bezeichnet werden soll (Abb. 6.6b). Zur übersichtlichen Darstellung einer solchen Merkmalserie, ist es notwendig sich geeigneter grafischer Mittel zu bedienen, die es ermöglichen, einen multidimensionalen Sachverhalt auf nur zwei Dimensionen abzubilden. Abbildung 6.6c zeigt ein Eigenschaftsprofil der Entwurfslösung A (Abb 6.5a) als Kreisdiagramm. Die Farben der Kreissegmente entsprechen jeweils dem erreichten Zielerfüllungsgrad eines Merkmals. Blau steht hierbei für einen hohen Wert, Rot für einen niederen. Eine hypothetische ideale Alternative, welche in allen Wertdimensionen die maximal erreichbaren Zielwerte erlangt, würde ein gänzlich blaues Profil bilden.<sup>9</sup>

<sup>9</sup> Vgl. Zangemeister 1970, 296.

## Fitnesswert einzelner Datenobjekte

Über die *gewichtete Summe* der ermittelten Zielerfüllungsgrade kann der *Fitnesswert* der Datenobjekte für Raumgruppen, Räume, Raumzonen oder räumliche Beziehungen innerhalb einer digital modellierten Entwurfslösung errechnet werden. Dieser Vorgang kann allgemein als *Wertsynthese* bezeichnet werden. Die einzelnen Zielerfüllungsgrade werden dabei als Teilziele betrachtet und zu einer gemeinsamen Zielfunktion zusammengefasst. Durch die Aggregation der Teilziele ist es möglich, ein multikriterielles Optimierungsproblem in ein monokriterielles abzubilden.

Sollen die einzelnen Teilziele  $f_1$  bis  $f_n$  unterschiedlich starken Einfluss auf die gemeinsame Zielfunktion haben, so können diesen unterschiedliche *Gewichtungsfaktoren*  $\omega_i$  als *Skalierungsparameter* zugeordnet werden. Die *gewichtete Summe* stellt in der Entscheidungstheorie das bekannteste und zugleich einfachste Verfahren zur rationalen Evaluierung von Varianten unter Einbeziehung mehrerer Teilziele dar. Ein hoher Wert als Gewicht bewirkt eine starke Veränderung des Zielfunktionswertes. Ein niedriger Wert bewirkt nur eine geringe Veränderung. Die Zielerfüllungsgrade einzelner Gebäudemerkmale sind auf das Intervall  $[0..1]$  normiert. Zur weiteren Verarbeitung ist es sinnvoll, dass auch die aggregierten Fitnesswerte  $f$  (Zielfunktionswerte) der Datenobjekte des Gebäudemodells im Intervall  $[0..1]$  liegen. Aus diesem Grund wird vereinbart, dass die Summe aller Gewichtungsfaktoren den Wert 1 ergeben muss.

$$f = \sum_{i=1}^k \omega_i f_i(x) \quad \text{wobei} \quad \sum_i \omega_i = 1$$

Die Zielerfüllungsgrade sowie die Gewichte bilden jeweils einen  $n$ -dimensionalen Vektor ( $v_s$  bzw.  $v_w$ ). Den gewichteten Zielfunktionswert erhält man durch die Bildung des *Skalarprodukts*. Die Koordinaten der Vektoren werden dazu paarweise multipliziert und die Produkte addiert. Das Ergebnis ist ein einfacher Zahlenwert (Skalar), der den Fitnesswert  $f$  des betrachteten Datenobjekts widerspiegelt.

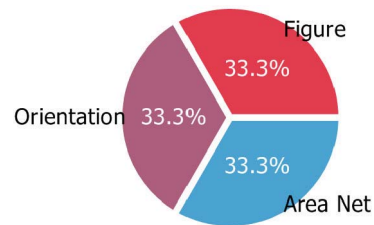
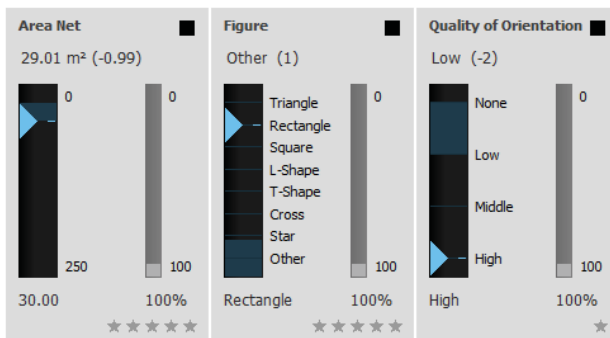
$$f = v_s \circ v_w = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \circ \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{pmatrix} \quad f(x) = f_1(x) \cdot \omega_1 + f_2(x) \cdot \omega_2 + \dots + f_n(x) \cdot \omega_n$$

Die einfachste Methode zur Definition der Kriteriengewichte stellt das *Direct Ratio Verfahren* dar. Dabei werden alle Gewichte über subjektive Schätzwerte zwischen 0 und 1 festgelegt. Soll die Gewichtung der Kriterien nicht einfach intuitiv festgesetzt werden, kann auch eine systematische Ermittlung der Prioritäten nach dem Prinzip des *gewichteten Paarvergleichs* erfolgen.<sup>10</sup>

<sup>10</sup> Beim paarweisen Vergleich aller Kriterien werden Punkte von 1 bis  $n$  vergeben. Der Eintrag  $n$  steht für eine absolute Dominanz eines Kriteriums gegenüber einem anderen. Der Wert 1 steht für die Gleichwertigkeit zweier

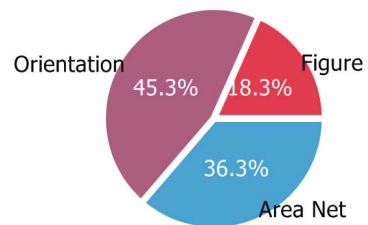
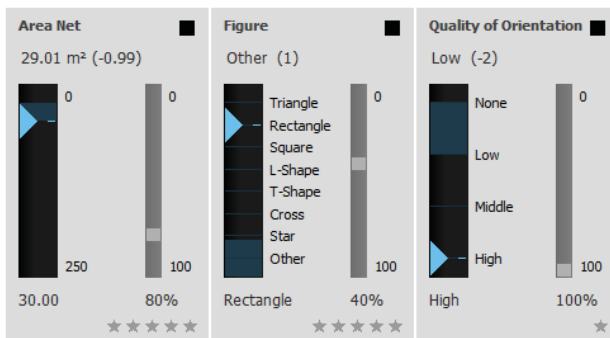
Beispielsweise kann für das Kriterium *Net Area* eine Priorität von 0.8, für *Quality of Orientation* 1.0 und für *Figure* 0.4 festgesetzt werden. Um der Forderung nachzukommen, dass alle Gewichtungswerte  $\omega_1$  bis  $\omega_n$  in Summe den Wert 1 ergeben, muss ein globaler Skalierungsfaktor berechnet werden. Diesen erhält man durch die Division von 1 durch die Summe aller Kriteriengewichte. Die Berechnung des Skalierungsfaktors im oben genannten Beispiel lautet:  $1 / (0.8 + 1.0 + 0.4)$  - dies ergibt 0,45. In der Folge werden die einzelnen Gewichte mit diesem Faktor multipliziert. Auf diese Weise errechnen sich die normierten Gewichtungswerte 0.363, 0.453 sowie 0.183, welche in Summe 1 ergeben. Aus diesen kann der normierte Gewichtungsvektor  $v_w$  gebildet werden. Abbildung 6.7 zeigt zwei unterschiedliche, im DESIGN GOAL MANAGER erstellte Prioritäten-Setups bei gleichmäßiger (a) sowie asymmetrischer Gewichtsverteilung (b). Mit dem Verschieben der einzelnen Sliders und somit der Prioritäten geht auch eine Veränderung des Fitnesswertes  $f$  einher, welcher das Skalarprodukt aus Gewichtungsvektor  $v_w$  und Satisfaktionsvektor  $v_s$  (6.6b) darstellt.

(a) gleichmäßige Gewichtsverteilung



$$f = v_s \circ v_w = \begin{pmatrix} 0.93 \\ 0.33 \\ 0.00 \end{pmatrix} \circ \begin{pmatrix} 0.333 \\ 0.333 \\ 0.333 \end{pmatrix} = 0.418$$

(b) asymmetrische Gewichtsverteilung



$$f = v_s \circ v_w = \begin{pmatrix} 0.93 \\ 0.33 \\ 0.00 \end{pmatrix} \circ \begin{pmatrix} 0.363 \\ 0.453 \\ 0.183 \end{pmatrix} = 0.487$$



Abb 6.7 Prioritäten-Mixer

Kriterien. Die restlichen Werte stehen für Nuancen zwischen diesen Extremen. Die aus den Paarvergleichen ermittelten Werte werden in eine Vergleichsmatrix übertragen, sodass dabei die Basiselemente (Reihen) mit den Vergleichselementen (Spalten) in Beziehung gesetzt werden. Die Evaluationsmatrix besteht dabei aus  $n \times n$  Elementen, wobei  $n$  gleich der Anzahl der Kriterien ist. Jedes Element der Matrix entspricht einem paarweisen Vergleich und somit einem Präferenzurteil. Diese Vorgehensweise ist beim *Analytic Hierarchy Process*, einer ähnlichen Methode wie die Nutzwertanalyse zur Unterstützung von Entscheidungsprozessen, sogar zwingend. Durch das paarweise Vergleichen der einzelnen Kriterien beziehungsweise der interessierenden Merkmale wird man dazu gezwungen, tiefgreifende Überlegungen zu deren Prioritäten anzustellen. Vgl. Mühlbacher u.a. 2013, 122f.

Die Fitness eines Datenobjekts im digitalen Gebäudemodell, dessen Merkmale unterschiedliche Wichtigkeit haben, kann durch ein Eigenschaftsprofil mit verschiedenen großen Kreissegmenten visualisiert werden. Die Winkel der Kreissegmente in Abbildung 6.7 sind jeweils proportional zu den Gewichtungsfaktoren der Kriterien. Die Farben der Segmente entsprechen den erreichten Zielerfüllungsgraden der interessierenden Merkmale. Kriterien mit höheren Gewichten bekommen dabei aufgrund der daraus resultierenden größeren Farbflächen automatisch mehr Dominanz. Das Rechenergebnis deckt sich mit dem Mittelwert der Farbgebung des jeweiligen Diagramms und kann deshalb auch intuitiv, auf rein visuellem Weg wahrgenommen werden.

## Gesamtfitness einer Entwurfslösung

Zur Ermittlung der Gesamtfitness einer Entwurfslösung, müssen die Fitnesswerte sämtlicher, in der Gebäudespezifikation beschriebener Objekte für Raumgruppen, Räume, Zonen und Raumbeziehungen im Gebäudemodell erfasst werden. Die Berechnung des Gesamtfitnesswertes kann danach über die erneute Bildung der gewichteten Summe dieser Werte erfolgen. Abbildung 6.8 zeigt eine Gebäudespezifikation und einen darauf basierenden Grundriss einer Entwurfslösung. Abbildung 6.9a zeigt die errechneten Fitnesswerte der einzelnen Räume und die entsprechend gefärbten Eigenschaftsprofile sowie die Fitnesswerte. Daneben in Abbildung 6.9b werden die einzelnen Fitnesswerte der einzelnen Raumobjekte als Vektor zusammengefasst. Die Gesamtfitness FIT der Entwurfslösung kann analog zur Berechnung der einzelnen Fitnesswerte (Abb. 6.7) durch das Skalarprodukt aus dem Fitnessvektor  $f$ , welcher sich aus Fitnesswerten der einzelnen Datenobjekte zusammensetzt und einem wiederum (frei) festgesetzten Gewichtungsvektor  $v_w$  berechnet werden. Bei gleichmäßiger Gewichtung von  $n$  spezifizierten Räumen ( $w_{1-n}=1/n$ ), entspricht die gewichtete Summe dem arithmetischen Mittel aller Fitnesswerte.

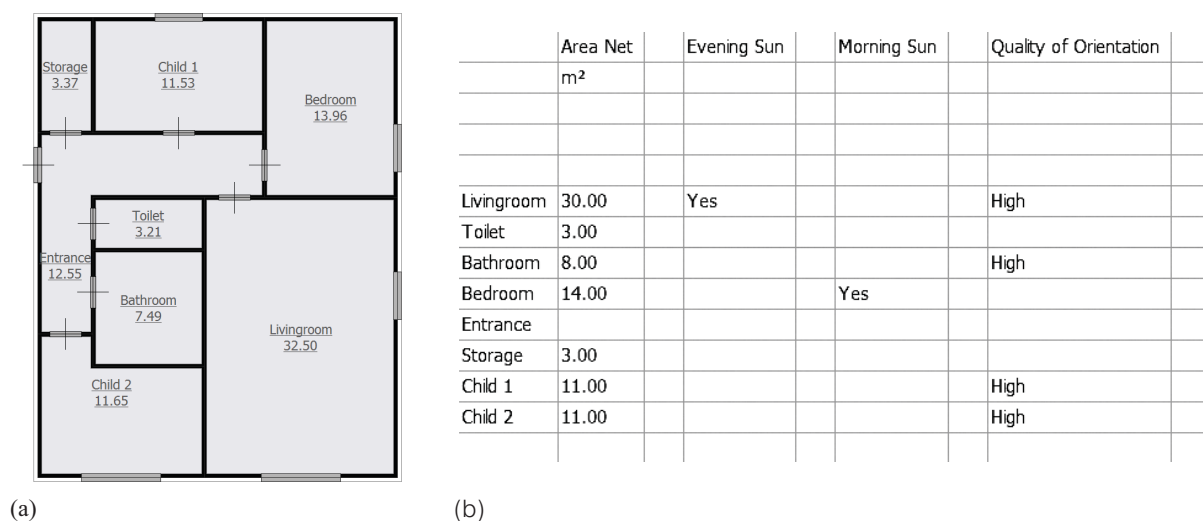


Abb 6.8 (a) Entwurfslösung (b) Spezifikation in Tabellenform



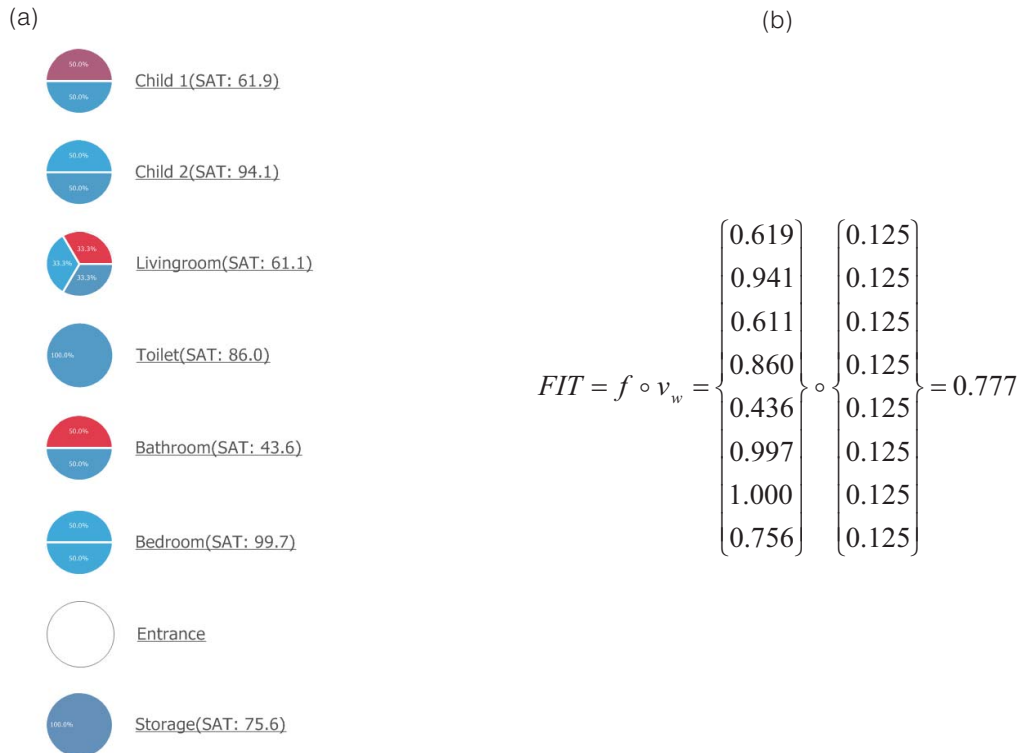


Abb 6.9 (a) Fitnesswerte, (b) Berechnung Gesamtfitness  $w_{1..n} = 1/8$ ,

Eine Besonderheit stellen in diesem Zusammenhang Räume oder andere Datenobjekte dar, welche in einer Gebäudespezifikation zwar benannt, jedoch durch keine besonderen Merkmale beschrieben werden. Dies ist im vorliegenden Beispiel bei dem Raum mit dem Namen *Entrance* der Fall. Für Funktionsbereiche wie diesen gilt, dass es nur wichtig ist, dass es sie gibt. Wird beispielsweise ein solcher „Raum ohne Eigenschaften“ im Datenmodell (Entwurf) identifiziert, wird diesem automatisch und unabhängig von seinen tatsächlichen Eigenschaften der Fitnesswert 1.0 (vor Gewichtung) zugewiesen.

Eine weitere Sonderstellung nehmen fehlende Räume ein. Wird ein Raum in einer Gebäudespezifikation genannt, aber im Datenmodell nicht gefunden, so vermindert dies in jedem Fall die Gesamtfitness der Entwurfslösung. Der Entwurf ist dann nicht vollständig. Bei gleichmäßiger Gewichtung kann dies rechnerisch durch eine prozentuelle Abminderung um  $1/n$  erfolgen, wobei  $n$  die Anzahl der in der Spezifikation beschriebenen Räume darstellt.

## Fitnessprofil

Die Visualisierung der Gesamtfitness einer Entwurfslösung kann über ein *Netzdiagramm*<sup>11</sup> erfolgen. Dazu werden die Fitnesswerte aller Datenobjekte strahlenförmig mit Hilfe von Polarkoordinaten in 360 Grad gleichmäßig um ein Zentrum angeordnet. Jedes Objekt erhält eine Achse, welche immer die gleiche Orientierung aufweist. Hohe Werte liegen wie bei einer Zielscheibe nahe am Zentrum. Danach werden die einzelnen Zielerfüllungsgrade (Intervall 0..1) jedes Datenobjekts auf den entsprechenden Achsen aufgetragen. Durch das Verbinden dieser Punkte erhält man einen geschlossenen Polygonzug, beziehungsweise bei entsprechender Füllung eine Fläche. Um eine Fläche aufspannen zu können müssen

<sup>11</sup> Netzdiagramme werden auch als Kiviat-, Radar- oder Sterndiagramme bezeichnet.

mindestens drei Datenobjekte (Merkmalsträger) existieren. Netzdiagramme eignen sich besonders gut zur Visualisierung und zum Vergleich von Merkmals-Serien, da diese den *Lösungsraum* sehr anschaulich wie ein Koordinatensystem um einen gemeinsamen Nullpunkt aufspannen (Abb 6.10). Die Anzahl der betrachteten Merkmale von Raumgruppen, Räumen, Zonen, und deren Beziehungen untereinander bildet die Dimension des Lösungsraums.

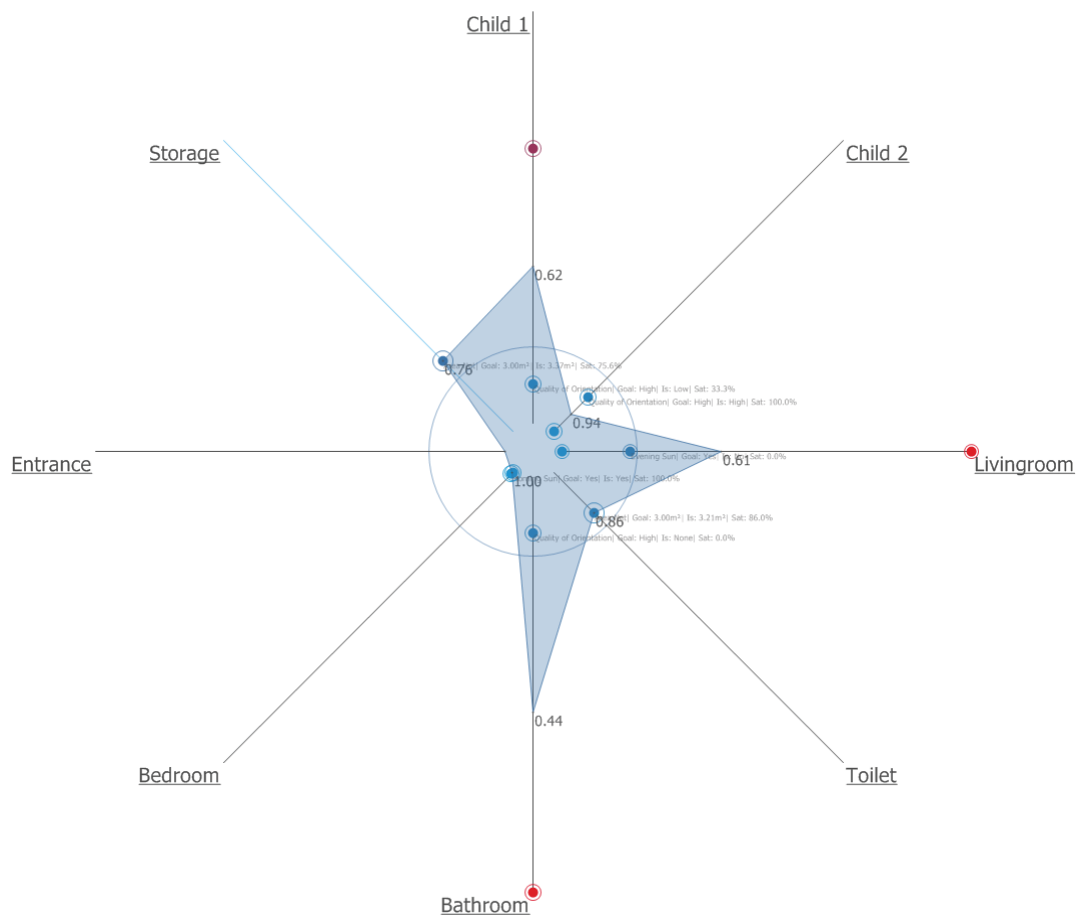


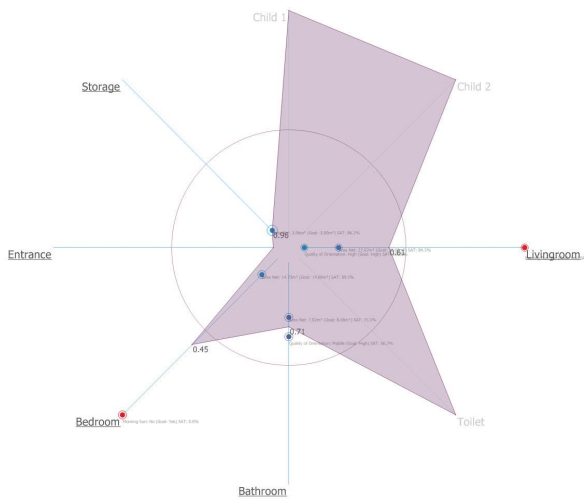
Abb 6.10 Fitnessprofil

## Vergleich unterschiedlicher Entwurfslösungen

Durch die Berechnung von Gesamtfitnesswerten ist es möglich, unterschiedliche Entwurfsvarianten systematisch nach ihrer Vorzugswürdigkeit, also nach der Höhe des Programmierfüllungsgrades zu ordnen. Auf diese Weise kann der Entscheidungsprozess zur Auswahl der passendsten Alternative unterstützt werden. Abbildung 6.11 zeigt vier Entwürfe der in Abbildung 6.8 spezifizierten Wohnung. In Abbildung 6.12 sind die dazugehörigen Fitnessprofile dargestellt. Sollen verschiedene Entwurfslösungen anhand einer Gebäudespezifikation detailliert miteinander verglichen werden, können die einzelnen Netzdiagramme überlagert werden. Durch die Überlagerung sind für einzelne Kriterien der Merkmalsserien vergleichende Aussagen möglich. Dennoch lassen sich solche mehrdimensionalen Mengen in der Regel als Ganzes nicht mehr vergleichen. Eine Ausnahme sind Mengen, die in allen Kriterien besser sind als eine Vergleichsmenge. Diese werden dann als pareto-optimal bezeichnet (vgl. Pareto-Prinzip: Kapitel 4.2, Entitäten). Im Netzdiagramm umschließt eine solche Menge die Vergleichsmenge(n) vollständig.



Abb 6.11 Vier Entwurfslösungen



(a) FIT: 46 %



(b) FIT: 77 %



(c) FIT: 57 %



(d) FIT: 73 %

Abb 6.12 Vier Fitnessprofile

## Problematik formalisierter Verfahren

Die mathematische Berechnung der Fitness einer architektonischen Entwurfslösung ist in jedem Fall kritisch zu betrachten. In mancher Hinsicht sind Verfahren wie die Nutzwertanalyse als Methode zur Auswahl einer optimalen Alternative nicht ganz unproblematisch. Aus diesem Grund soll nicht darauf verzichtet werden, neben den positiven Aspekten und Vorteilen auch auf deren Grenzen hinzuweisen:

Zum einen handelt es sich hierbei um eine kompensatorische Methode. Dies bedeutet, dass Vor- und Nachteile einzelner Varianten, die miteinander verglichen werden, gegeneinander verrechnet werden. Vorteile bezüglich eines Kriteriums können dadurch Nachteile in Bezug auf ein anderes kompensieren.<sup>12</sup> Bei der Berechnung der Fitness einer Entwurfslösung werden sehr komplexe und differenzierte Sachverhalte auf nur eine Zahl reduziert. Die unterschiedlichen Zielerfüllungsfaktoren sind in Anbetracht des Gesamtfitnesswertes nicht mehr nachvollziehbar. Das arithmetische Endergebnis ist als solches nicht anschaulich und kann nur durch ein differenziertes Monitoring der einzelnen Aggregationsschritte nachvollziehbar gemacht werden.

Zum anderen werden bei der Berechnung der Fitness Rechenoperationen angewandt, die eigentlich nur für Merkmalsausprägungen zulässig sind, welche auf kardinalen Skalenniveaus quantitativ erfasst werden. Im Zuge der Datenverarbeitung werden den Ausprägungen qualitativer Merkmale auf der Ordinalskala (gute, durchschnittliche, schlechte Orientierung) Zahlen (1, 2, 3) zugewiesen. Durch die Zuordnung einer Zahl zu jeder Kategorie erfolgt eine Kodierung der Wertemenge. Diese Zahlen stellen jedoch im Grunde nur Stellvertreter vager verbaler Beschreibungen von Nuancen dar. Dies kann leicht zu einer Scheingenauigkeit führen. Es darf nicht außer Betracht gelassen werden, wie viel Unsicherheit der Information sich hinter diesen Werten verbergen kann.<sup>13</sup> Auch bringt die Methode der Nutzwertanalyse mit sich, dass es bei lückenhaften Zielanalysen sowie methodisch unzureichenden Gewichtungen mitunter zu Verfälschung der Ergebnisse kommen kann.<sup>14</sup> Die numerischen Gesamtergebnisse (Fitnesswerte) dürfen auf keinen Fall überinterpretiert werden. Darüber hinaus besteht in formalisierten Verfahren, häufig die Neigung, leicht Messbares sehr detailliert zu erfassen und zu verarbeiten und schwierig erfassbare Kriterien eher zu vernachlässigen. Dies kann zu einem Übergewicht der harten Daten im Entscheidungsprozess führen.<sup>15</sup> Aus den genannten Gründen darf eine ausschließlich maschinelle Evaluierung architektonischer Entwurfslösungen anhand gänzlich formalisierter Kriterien in keinem Fall unreflektiert erfolgen.

## Zusammenfassung

Durch die Bildung einer abstrakten Objektklasse können die ARCHITEKTUR ROUTINE(n) in Computersysteme eingebunden werden. Durch die Berechnung und grafische Darstellung der Zielerfüllungsgrade sämtlicher, in einer Gebäudespezifikation beschriebener und in einem Gebäudedatenmodell erhobener Teilziele einer Entwurfsaufgabe, kann eine spektrale

---

<sup>12</sup> Vgl. Müller-Herbers 2007, 50.

<sup>13</sup> Vgl. Fahrmeir 2011, 17.

<sup>14</sup> Vgl. Müller-Herbers 2007, 41.

<sup>15</sup> Vgl. Müller-Herbers 2007, 50.

Analyse von Entwurfslösungen in Bezug auf eine endliche Anzahl interessierender Kriterien erfolgen. Die Nutzwertanalyse stellt ein transparentes und nachvollziehbares Mittel zur Unterstützung der Entscheidung zwischen mehreren, vorliegenden Entwurfsvarianten dar. Im Zuge der Gewichtung der Kriterien erfordert die Anwendung dieser Methode bei der Evaluierung von Gebäudeentwürfen die genaue Überprüfung der Wichtigkeit jedes einzelnen Gebäudemerkmals (Kriteriums). Dies kann unter Umständen schon vor dem eigentlichen Entscheidungsprozess zu einem wertvollen Erkenntnisgewinn führen. Aufgrund der Zahlendarstellungen wird eine rationale Form der Vergleichbarkeit (selbst subjektive Faktoren) hergestellt, die ohne diese Methode nicht gegeben ist. Bei der Nutzwertanalyse erfolgt eine Bilanzierung indem alle Argumente für oder gegen die verschiedenen Entwurfslösungen gegenübergestellt und mit Gewichtungsfaktoren versehen werden. Die Aggregation der gewichteten Kriterien ergibt ein mathematisch eindeutiges Ergebnis, welches ein sehr komplexes rückführbares (deduktives) Argument darstellt. Um die multidimensionalen Evaluierungsergebnisse auch sinnvoll nutzen zu können, stellt dabei die Wahl der adäquater Visualisierungsmethoden sowie die Entwicklung geeigneter grafischer User Interfaces eine besondere Herausforderung dar.

## 6.2 Neue Instrumente für den Architektorentwurf

Das folgende Kapitel beschreibt anhand der vom Autor konzipierten Software ARCHILL.ES, wie die Potenziale der ARCHITEKTUR ROUTINE(n), sowie das vorgestellte Konzept einer maschinellen Evaluierung von Entwurfslösungen, genutzt und der Praxis des Entwerfens zugänglich gemacht werden kann. Es wird gezeigt, wie dadurch die Arbeitsbedingungen des architektonischen Entwerfens verändert und verbessert werden können.

Wenn Anwendungsprogramme objektorientiert implementiert werden, ist es möglich aus diesen Objekten wiederum neue Softwaresysteme zusammenzusetzen. Dieser Vorteil der objektorientierten Programmierung wird bei der Entwicklung von ARCHILL.ES genutzt. ARCHILL.ES besteht aus der Vereinigung des Building Information Tools DRAFTPAD (Kapitel 4.2) und dem DESIGN GOAL MANAGER (Kapitel 3.5) zu einer architektonischen Entwurfsumgebung. Das dritte und verbindende Element stellt die Sammlung der ARCHITEKTUR ROUTINE(n) dar, welche die ursprünglich getrennten Komponenten wechselseitig miteinander in Beziehung setzen (Abb 6.14). Mit Hilfe von ARCHILL.ES können Gebäude spezifiziert, entworfen und evaluiert werden.

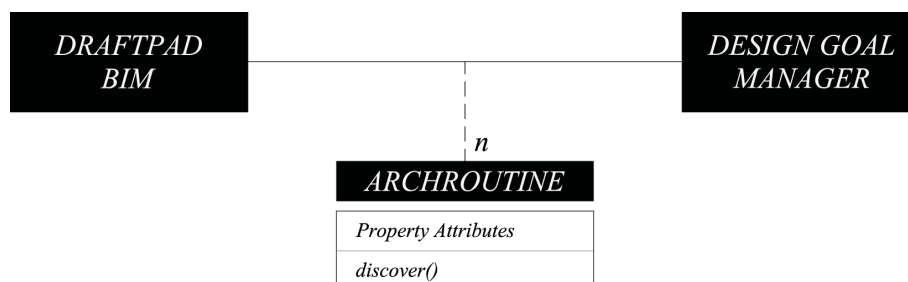


Abb 6.13 ARCHILL.ES Programmkomponenten



(a)

(b)

Abb 6.14 Martin Emmerer 2014, DRAFTPAD und DESIGN GOAL MANAGER

Auf der einen Seite können im DESIGNGOAL MANAGER, dem grafischen Editor zur Erstellung von differenzierten Gebäudespezifikationen, Entwurfsaufgaben parametrisch formuliert werden. Die Benutzerin oder der Benutzer modelliert die topologische Struktur der Entwurfsaufgabe als Graph, dessen Knoten und Kanten durch diverse Wunschmerkmale gewichtet werden (Abb 6.14b). Dies erfolgt durch das Andocken von „Merkmalsmodulen“. Die Bedeutung der vom Programm zur Verfügung gestellten Raum- und Beziehungsmerkmale wird durch einen Kurztext explizit erklärt. Dieser beinhaltet nicht nur eine exakte begriffliche Definition (Bedeutungsanalyse), sondern erklärt auch wie und auf welchem Skalenniveau das Merkmal im räumlichen Entwurf erfasst wird. Das differenzierte Setup der Variablen erfolgt über ein spezielles Mischpult, auf dem die gewünschte Ausprägung jedes Merkmals von der Benutzerin oder vom Benutzer über Schieberegler eingestellt wird. Die Skalen qualitativer Merkmale setzen sich aus begrifflichen Kategorien zusammen, während Schieberegler für quantitative Merkmale numerische Intervalle repräsentieren. Während das Merkmalsmodul *Orientierung* beispielsweise nur die Wahl zwischen „sehr gut“, „gut“ und „schlecht“ erlaubt, kann beim Modul *Raumgröße* die gewünschte Zahl der Quadratmeter durch Verschieben des Reglers beliebig festgesetzt werden.

Auf der anderen Seite können im Zeichenprogramm DRAFTPAD räumliche Entwürfe geometrisch entwickelt werden (Abb. 6.14a). Hierfür stehen zahlreiche speziell auf die Praxis des Entwerfens zugeschnittene Werkzeuge zur Transformation räumlicher Arrangements zur Verfügung. Die Konturen der gewünschten Räume werden gezeichnet und auf einer oder mehreren Ebenen angeordnet. Ebenso können Elemente wie Türen, Fenster, vertikale Erschließungen etc. platziert werden.

## Wechselspiel

Die Koppelung der beiden Komponenten DRAFTPAD und DESIGN GOAL MANAGER erlaubt verschiedenste Arbeitsweisen. Zum einen ist es möglich, ein differenziertes Gebäude-setup als Graph zu erstellen oder aus einer Tabelle zu laden und danach auf Knopfdruck in das Programm DRAFTPAD zu transferieren. Die Eigenschaft der Nettogröße kann hierbei direkt in die Geometrie übernommen werden, indem automatisch maßstäbliche Quadrate mit den exakten Nettoflächen gebildet werden. Dieser Vorgang ist in der Praxis des Entwerfens einer der ersten Tätigkeiten in der frühen Entwurfsphase. Abbildung 6.15 zeigt dies am Beispiel eines einfachen Raumprogramms.

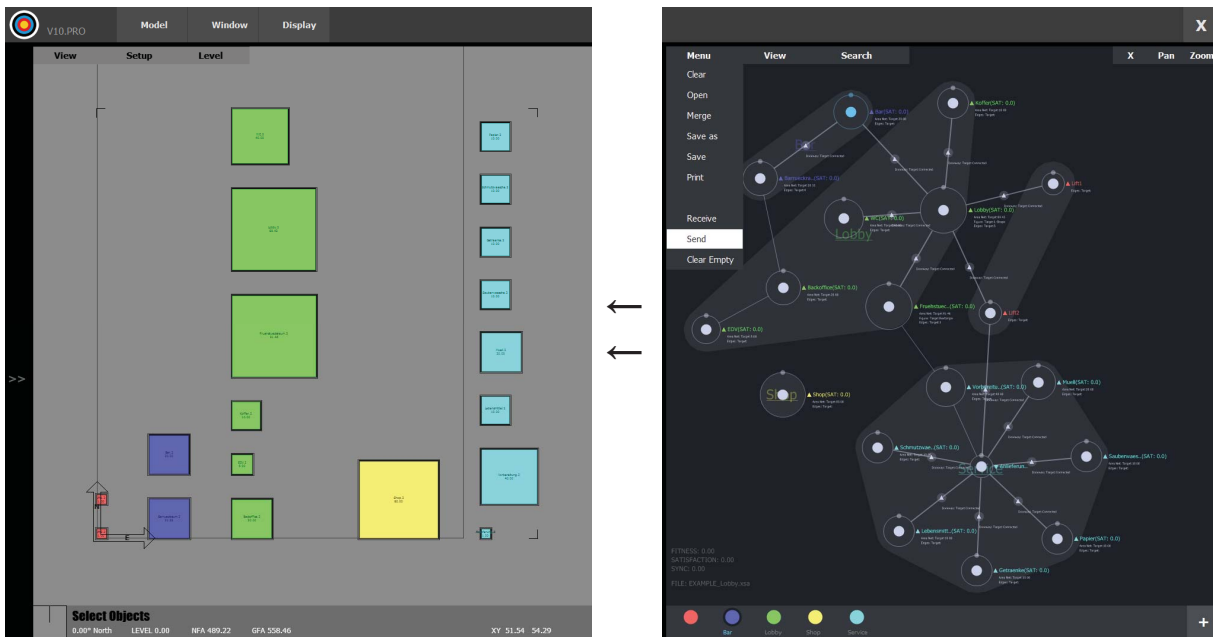


Abb. 6.15 ARCHILL.ES: Von der Topologie zur Geometrie

Genauso gut kann von der Geometrie ausgegangen werden. Dazu wird ein Gebäudeentwurf zuerst geometrisch in der Komponente DRAFTPAD entwickelt oder aus einer Datei geladen. Daraufhin kann davon automatisch eine topologische Gebäudespezifikation abgeleitet werden (Abb. 6.16). Hierzu werden die Nachbarschaften und Zugänglichkeiten im Building Information Model ermittelt und der duale Graph abgeleitet. Die Knoten und Kanten des Graphen werden mit den interessierenden, im Gebäudedatenmodell durch ARCHITEK-

TUR ROUTINE(n) erfassten Merkmalsausprägungen als Gewichte versehen. Auf diese Weise kann eine Art topologischer Fingerprint eines geometrischen Gebäudeentwurfs generiert werden (Abb 6.16). Ein Anwendungsfall für diese Funktionalität ist beispielsweise die Ableitung einer Gebäudespezifikation aus einem musterhaften Entwurf (z.B. Leitprojekt eines Wettbewerbes o.ä.). Da es sich dabei um eine explizite Beschreibung einer Entwurflösung handelt, kann ein solcher Fingerprint auch zu Argumentationszwecken benutzt werden, um die Qualitäten eines Entwurfs faktisch darzulegen.

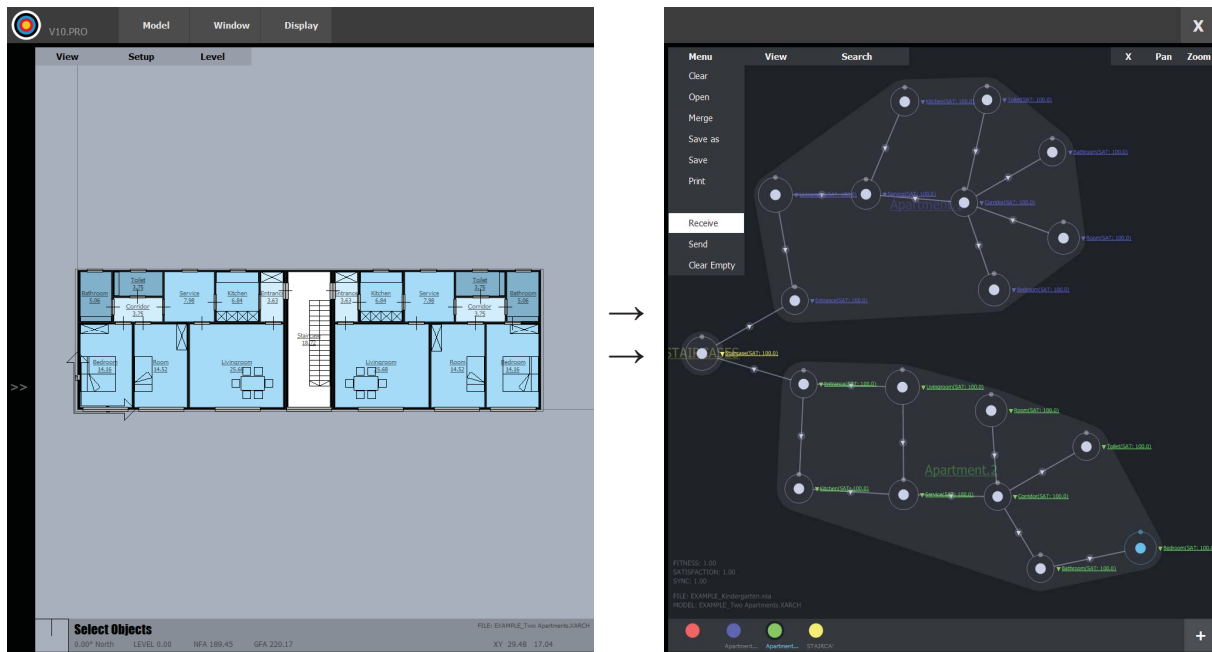


Abb 6.16 ARCHILL.ES: Von der Geometrie zur Topologie

## Parallele Betrachtung von Topologie und Geometrie

Der geometrisch konstruierte, räumliche Entwurf und die topologisch modellierte Struktur eines Gebäudes bestehen bei ARCHILL.ES parallel nebeneinander. Die Knotenpunkte des Graphen werden über die Raumbezeichnungen mit ihren geometrischen Repräsentanten im Entwurf synchronisiert. Jedem (bezeichnenden) Knoten wird bei diesem Suchvorgang das bezeichnete Datenobjekt als Pointer zugewiesen. Im Hintergrund arbeiten die Entdeckungsalgorithmen der ARCHITEKTUR ROUTINE(n), welche die Abweichungen von der im DESIGN GOAL MANAGER formulierten Gebäudebeschreibung berechnen.

Das System ist so in der Lage, ein differenziertes Feedback darüber zu geben, ob und zu welchem Grad eine Entwurflösung die ihr zugrundeliegende Gebäudespezifikation erfüllt. Mathematisch betrachtet erzeugt dies ‚Stress im Lösungsraum‘.<sup>16</sup> Nach der Aktivierung des sogenannten *Stressmodes*, werden die Abweichungen durch Farben dargestellt.

<sup>16</sup> Der Ausdruck *Lösungsraum* bezeichnet hier die Menge der möglichen Entwurflösungen. Mit Stress ist in diesem Zusammenhang der Optimierungs-Druck gemeint, welcher mit dem Wunsch nach einer möglichst guten Lösung einhergeht. In Optimierungsmodellen sinkt der Stress mit dem Ansteigen des Optimierungsgrades.

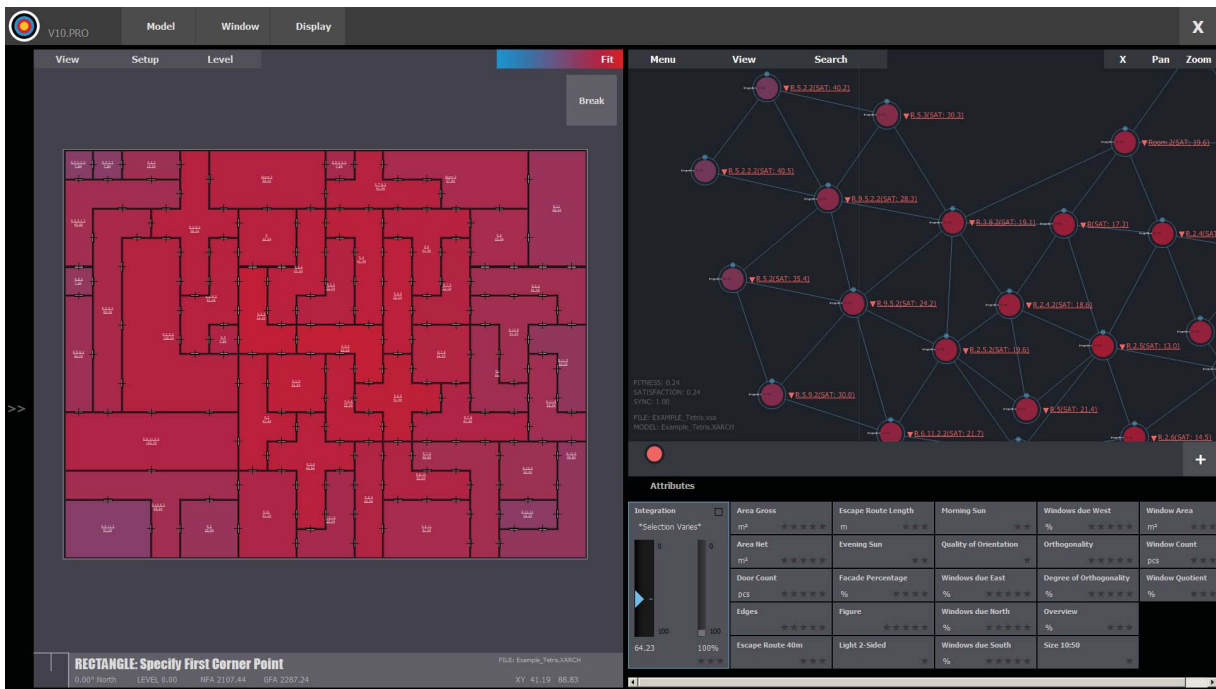


Die Abbildungen 6.17a, b und c zeigen beispielhaft das farbige Feedback in Bezug auf Orthogonalität, vordefinierte Nettoraumgrößen und den aus Space Syntax stammenden Analyseparameter *Integration*<sup>17</sup>.



(a)

(b)



(c)

Abb 6.17 Screenshots ARCHILL.ES: (a) Orthogonalität (b) Nettonutzflächen (c) Integration

<sup>17</sup> Vgl. Hillier/Hanson 1984.

Abbildung 6.18 zeigt ein weiteres Anwendungsbeispiel: Vier Varianten derselben Wohnanlage. Für jede Wohneinheit wurde der Status „gut orientiert“ definiert. Der Plan ist genordet. Man kann sehen, wie sich die Qualität der Orientierung der einzelnen Wohneinheiten durch die Rotation des Gesamtbaukörpers um jeweils 45% verändert. Die Farbnuancen geben Auskunft über den Grad der Erfüllung des topologisch formulierten Zustandes. Dabei steht Blau für sehr gute, und Rot für eher schlechte Orientierung. Gemäß den Definitionen der ARCHITEKTUR ROUTINE *Quality of Orientation* (siehe Anhang) wird die Kategorie „high“ nur an Wohnungen vergeben, welche nach Süden oder Westen orientiert sind, „medium“ bedeutet eine Orientierung nach Osten und „low“ bezeichnet ausschließlich nach Norden orientierte Wohneinheiten. Die farbige Visualisierung stellt ein eindeutiges und unmittelbares Feedback dar und erlaubt es der Entwerferin oder dem Entwerfer, intuitiv und planerisch gezielt auf mögliche Schwachpunkte des Entwurfes zu reagieren, beziehungsweise die beste Variante auszuwählen.

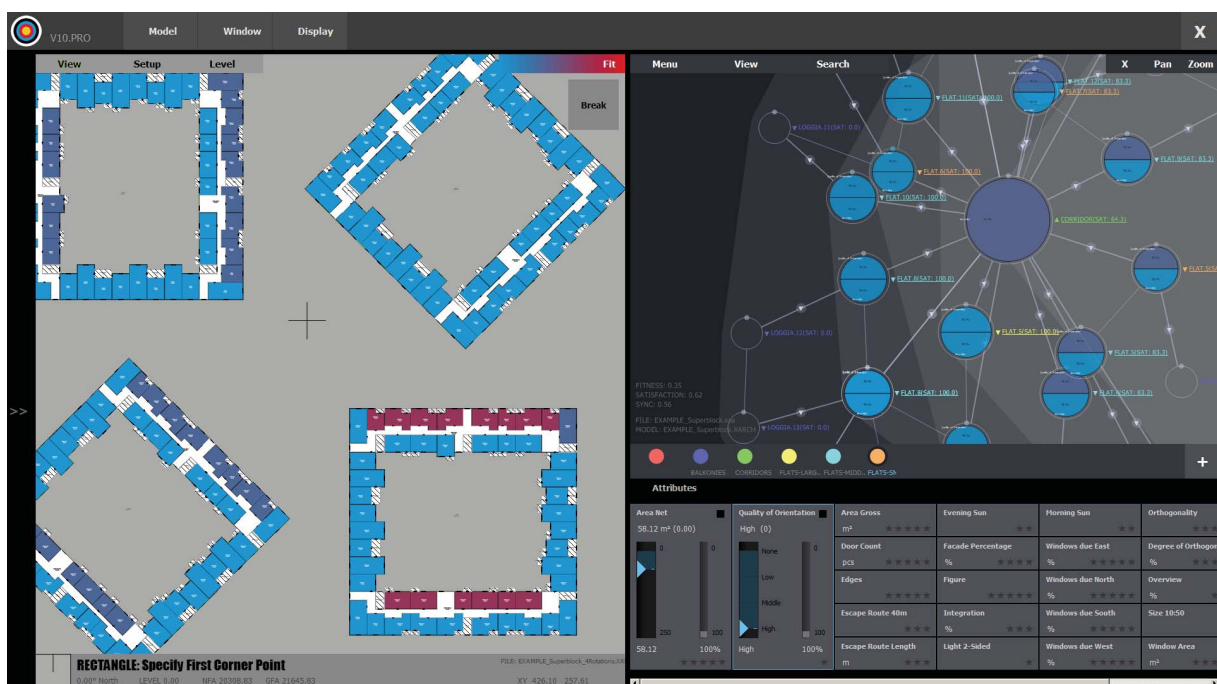
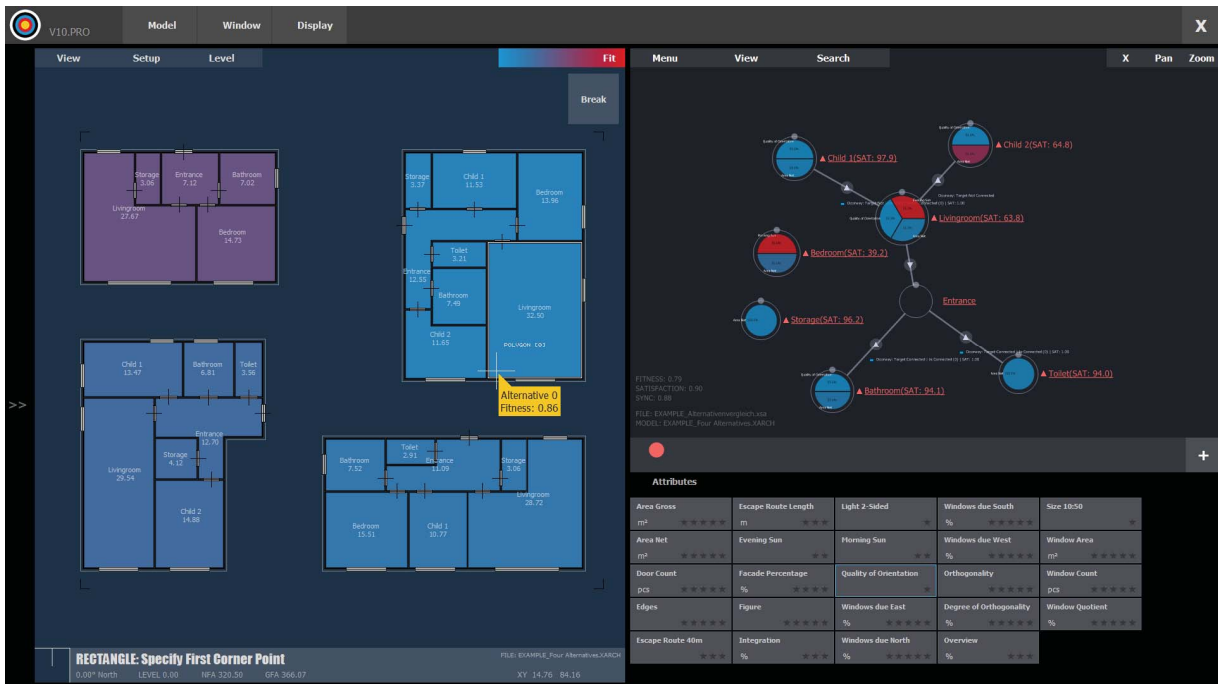
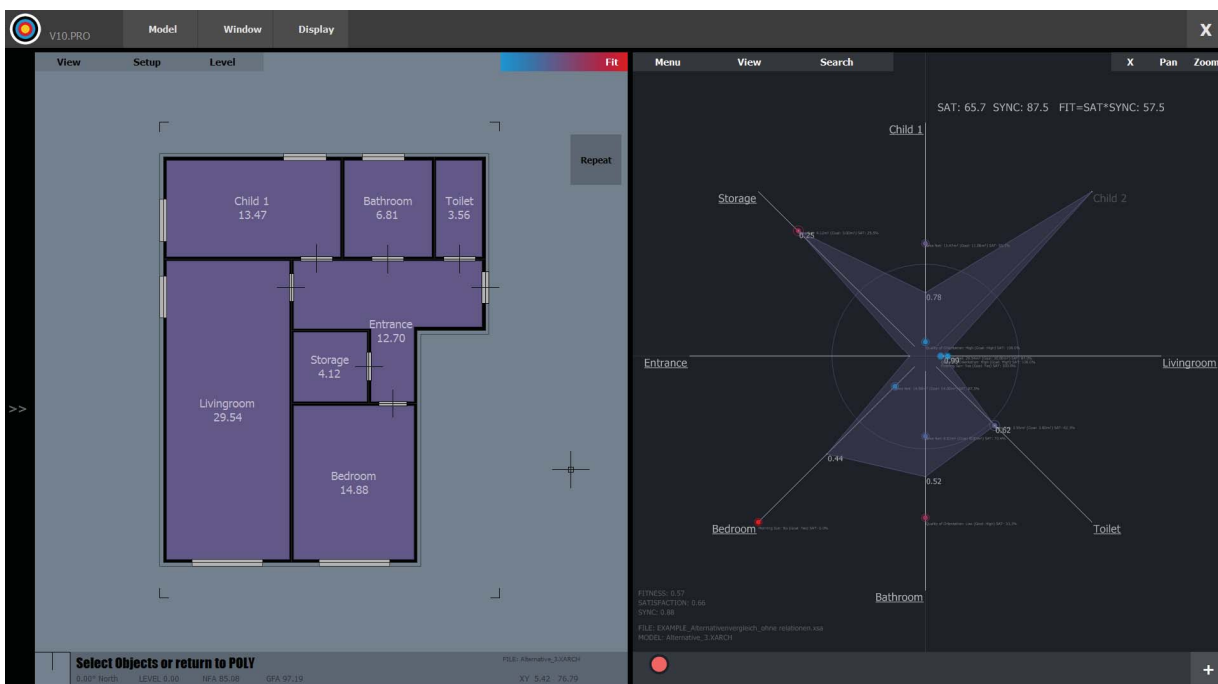


Abb 6.18 Screenshot ARCHILL.ES: Qualität der Orientierung von Wohneinheiten

Die Abbildungen 6.19 a und b zeigen den Vorgang der Evaluierung ganzer Entwurflösungen mit Hilfe des Softwaretools ARCHILL.ES. Screenshot a zeigt eine als Graph modellierte Spezifikation einer Wohnung. Daneben, auf der linken Seite sind vier unterschiedliche Entwurflösungen zu sehen. Die Farbe jedes Grundrisses entspricht dem jeweiligen Fitnesswert bezüglich der in Abbildung 6.8 dargestellten Spezifikation, auf einer Farbskala von Rot für geringe Fitness bis Blau für hohe Fitness. Screenshot b zeigt eine Entwurflösung im Detail mit ihrem entsprechenden Fitnessprofil.



(a)



(b)

Abb 6.19 Screenshots ARCHILL.ES:

(a) Gebäudespezifikation als Graph und unterschiedlicher Entwurflösungen (b) Entwurf und Fitnessprofil

Abbildung 6.20 zeigt eine weitere Funktion des Computerprogramms ARCHILL.ES: Die facettierten Suche nach Räumen (oder anderen Datenobjekten) mit bestimmten Eigenschaften. Der DESIGN GOAL MANAGER wird dabei zur Suchmaschine. Auf der rechten Seite wird ein differenziertes Raum Setup gezeigt. Die gewünschten Raumqualitäten werden durch die ARCHITEKTUR ROUTINE(n) *Area Net*, *Quality of Orientation*, *Light 2-sided* und *Facade Percentage* beschrieben. Die unterschiedlich großen Abschnitte des Kreisdiagramms zeigen die festgelegten Prioritäten der verschiedenen Merkmale.

Auf der linken Seite ist in der Komponente DRAFTPAD der Grundriss einer Wohnanlage mit unterschiedlichsten Wohnungstypen zu sehen. Mit jeder Veränderung eines Sliders und somit der Suchkriterien, wird über die im Hintergrund arbeitenden ARCHITEKTUR ROUTINE(n) ein Suchergebnis generiert. Jene Treffer, welche den anhand der ARCHITEKTUR ROUTINE(n) über Schieberegler definierten Suchkriterien am ehesten entsprechen, die sogenannten *Best Matching Units*, werden hellblau gekennzeichnet. Je weiter die Ergebnisse von dem parametrischen Setup abweichen, desto mehr färbt sich der Raum rot. Bei jeder Veränderung der Zielwerte und Gewichtungen ändern sich die Suchergebnisse und damit das Farbmuster in Echtzeit. Dasselbe funktioniert auch mit ganzen Entwurfsvarianten oder Teilen davon. Die Eingabe über die Slider des GOAL MANAGERS ist komfortabel. Alternativ kann die Eingabe der durch ARCHITEKTUR ROUTINE(n) operationalisierten Begriffe, wie bei Suchmaschinen im Internet über ein Textfeld erfolgen.

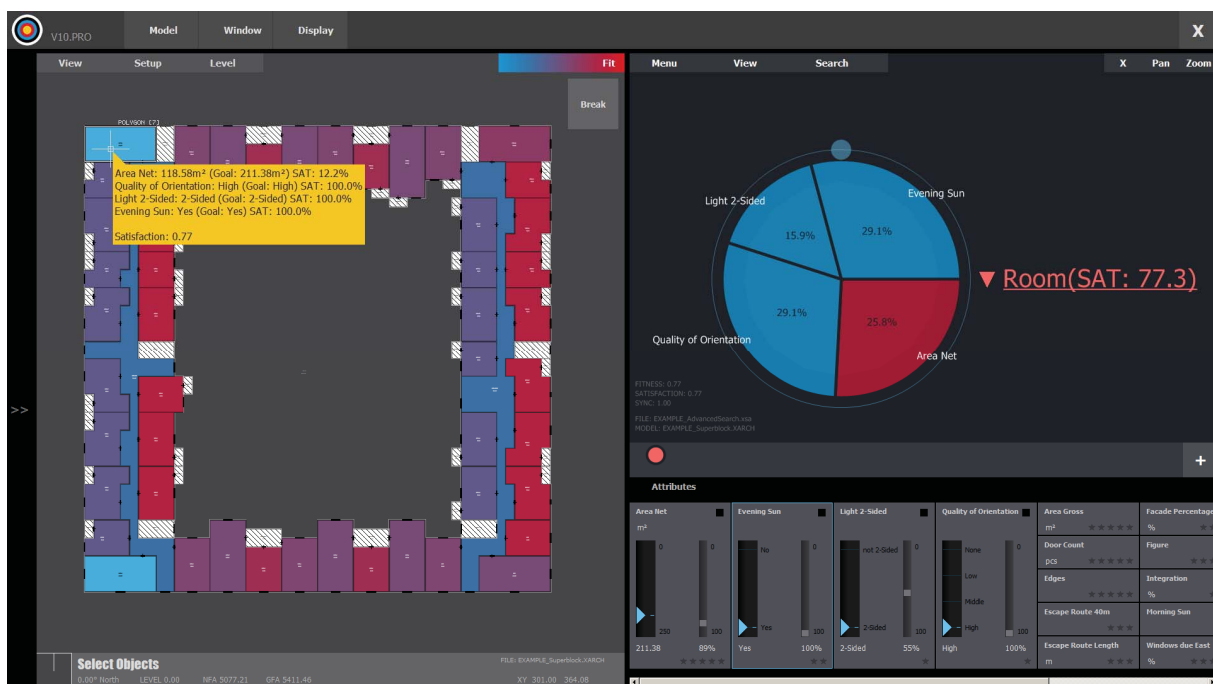


Abb.6.20 Screenshot ARCHILL.ES: Facettierte Suche im Building Information Model

## Zusammenfassung

Das Grundkonzept von ARCHILL.ES beruht auf dem symmetrischen Nebeneinander der topologischen Beschreibung eines Gebäudes (Genotyp) und einer geometrisch konstruierten Entwurfslösung (Phänotyp). Die räumliche Entwicklung eines Gebäudes kann in DRAFT-PAD wie gewohnt manuell erfolgen. Zur Ausführung dieser Tätigkeit wird der Entwerferin oder dem Entwerfer eine „freundliche“ Umgebung nach Robin M. Hogarth (vgl. Einleitung S.2) geboten. Hierzu werden die Potenziale des Computers genutzt. Die Funktionalitäten der ARCHITEKTUR ROUTINE(n) können durch die visuelle Entwicklungsumgebung ARCHILL.ES auf unterschiedliche Weise genutzt werden. Während die Entwerferin oder der Entwerfer das räumliche Arrangement über ein einfach zu bedienendes, grafisches Interface manipuliert, laufen im Hintergrund parallel dazu die Algorithmen der ARCHITEKTUR ROUTINE(n), welche diesen Vorgang überwachen und ständig ein differenziertes Feedback darüber geben, ob und wie weit die Zielvorstellungen bereits erfüllt sind. Die Hinweise erfolgen sowohl visuell als auch sprachlich.<sup>18</sup> Das Programm erlaubt sowohl die Berechnung der Gesamtfitness einer oder mehrerer Entwurfslösungen als auch eine facettierte Suche innerhalb eines Building Information Models.

## 6.3 Schlussfolgerungen

Werden die Anforderungen an ein zukünftiges Gebäude in Form einer XSAL-notierten Gebäudespezifikation durch Merkmale beschrieben, welche auch als ARCHITEKTUR ROUTINE(n) operationalisiert wurden, können die Abweichungen von den einzelnen Zielvorstellungen (Satisfaktionsvektor) mit Hilfe von Präferenzfunktionen maschinell ermittelt werden. Durch Eigenschaftsprofile und Netzdiagramme kann ein anschauliches Monitoring der Entwurfslösung erfolgen. Die Aggregation der einzelnen (Teil-)Zielerfüllungsgrade nach dem Prinzip der Nutzwertanalyse ermöglicht die Bildung der Gesamtfitnessfunktion (Objective Function) einer spezifischen Entwurfsaufgabe. Mit Hilfe dieser Funktion, kann in der Folge die Fitness einer oder mehrerer Entwurfslösungen berechnet werden. Dadurch kann beim Vorliegen mehrerer Alternativen eine Reihung vorgenommen, beziehungsweise die Entscheidung für eine Entwurfsvariante unterstützt und mathematisch transparent belegt werden. Das User Interface des DESIGN GOAL MANAGERS ermöglicht die intuitiv Durchführung dieses mathematisch relativ komplexe Vorgangs ohne jede Programmierkenntnisse.

Durch Tools wie ARCHILL.ES können die Potentiale der ARCHITEKTUR ROUTINE(n), sowie der vorgestellten Evaluierungsmethode, der Praxis des Entwerfens benutzerfreundlich zugänglich gemacht werden. Das begleitende Feedback identifiziert Schwachpunkte von Gebäudeentwürfen bereits während des Entwurfsprozesses, wodurch geeignete Schritte zur Verbesserung des räumlichen Arrangements gesetzt werden können.

---

<sup>18</sup> Vgl. Emmerer 2014. 63.

## CONCLUSION – DER KREIS SCHLIEßT SICH

*»Some people consider it noble to have some method, others consider it equally noble to have no method. To have no method is bad. To adhere strictly to method is worse still. It is necessary at first to observe a strict rule, then to penetrate intelligently into all the transformations. The possession of method liberates us from the necessity of possessing method.«<sup>1</sup>*

Lao-Tse, 6.Jhdt.v.Chr.

Die Errungenschaft der vorliegenden Arbeit liegt, im übertragenen Sinn, in einer *Alphabetisierung*<sup>2</sup> des Computers. Der Maschine wurden zwei neue Fähigkeiten beigebracht - einerseits das Lesen von Raumprogrammen und auf der anderen Seite das beziehungsweise Erkennen von Qualitäten einer Entwurfslösung in einem Building Information Model.

Ersteres wird durch die Formulierung von Gebäudespezifikationen nach dem Standard der EXPANSIBLE SPACE ALLOCATION LANGUAGE (XSAL) vorbereitet. Bei der Datenverarbeitung erfolgt eine Umwandlung der, in dieser formalen Sprache beschriebenen Gebäudemerkmale in einen n-dimensionalen Vektor  $v_p$ , wobei n der Anzahl der im Programm formulierten Zielvorstellungen entspricht. Abstrakt und aus dem Blickwinkel des *Operations Research* betrachtet, erzeugt die Gebäudespezifikation durch diesen Vektor *Stress* im zuvor uninformatierten Lösungsraum. Auf der anderen Seite machen die ARCHITEKTUR ROUTINE(n) Entwurfslösungen maschinenlesbar. Voraussetzung hierfür wiederum ist die Modellierung in einem dafür geeigneten Gebäudedatenmodell (vgl. Kapitel 4.2, DRAFTPAD). Die Ergebnisse der Entdeckungsalgorithmen, das heißt die erfassten Ausprägungen der interessierenden Merkmale in einer Entwurfslösung bilden ihrerseits auch einen Vektor  $v_D$ , welcher ebenfalls n Dimensionen aufweist. Die möglichen Anwendungen dieser Lesarten sind vielfältig.

Beim Durchlaufen der ALGORITHMISCHEN EXPLIKATION werden Begriffe, welche in der Alltagssprache zur Beschreibung von Gebäudemerkmalen benutzt werden, segmentiert (Bedeutungsanalyse), und daraufhin neu konstruiert (Operationalisierung).

*»Der strukturelle Mensch nimmt das Gegebene, zerlegt es, setzt es wieder zusammen; das ist scheinbar wenig (und veranlasst manche Leute zu der Behauptung, die strukturalistische Arbeit sei, unbedeutend, uninteressant, unnützlich usw.). Und doch ist dieses Wenige, von einem anderen Standpunkt aus gesehen, entscheidend; denn zwischen den beiden Objekten, oder zwischen den beiden Momenten strukturalistischer Tätigkeit, bildet sich etwas Neues, und dieses Neue ist nichts Geringeres als das allgemeine Intelligible<sup>3</sup>: das Simulacrum, das ist der dem Objekt hinzugefügte Intellekt, und dieser Zusatz hat insofern ei-*

<sup>1</sup> Lao-Tse, zit. n. Murphy 2005, S.49.

<sup>2</sup> Überwindung des Analphabetismus.

<sup>3</sup> „nur durch den Intellekt und nicht durch die sinnliche Wahrnehmung erkennbar.“ Duden online, intelligibel, 20.10.2014.

*nen anthropologischen Wert, als er der Mensch selbst ist, seine Geschichte, seine Situation, seine Freiheit und der Widerstand, den die Natur seinem Geist entgegengesetzt.»<sup>4</sup>*

Der Prozess der ALGORITHMISCHEN EXPLIKATION beinhaltet eine zweifache Translationsleistung. Auf der einen Seite führt dieser Vorgang zu einer intersubjektiven Klärung der Begrifflichkeiten, welche bei der Beschreibung von Gebäuden verwendet werden. Es wird beispielsweise festgesetzt was „gute Orientierung“ eines Wohnraums bedeutet und auf welchen Kriterien eine solche Beurteilung beruht. Andererseits erfolgt die exakte Definition der Handlungsschritte, welche zur Identifikation (zum maschinellen Erkennen) dieser Qualität in einer digital modellierten Entwurfslösung vollzogen werden müssen.

Durch die Implementierung dieses Vorgangs in einen Algorithmus scheidet der Mensch selbst aus dem Evaluierungsprozess aus. Das Programm kann aber von der Maschine beliebig oft und auf beliebige Objekte angewendet werden. Gemäß den Prinzipien wissenschaftlicher Experimente wird der persönliche Standpunkt der Forscherin oder des Forschers auf diese Weise weitestgehend entsubjektiviert.

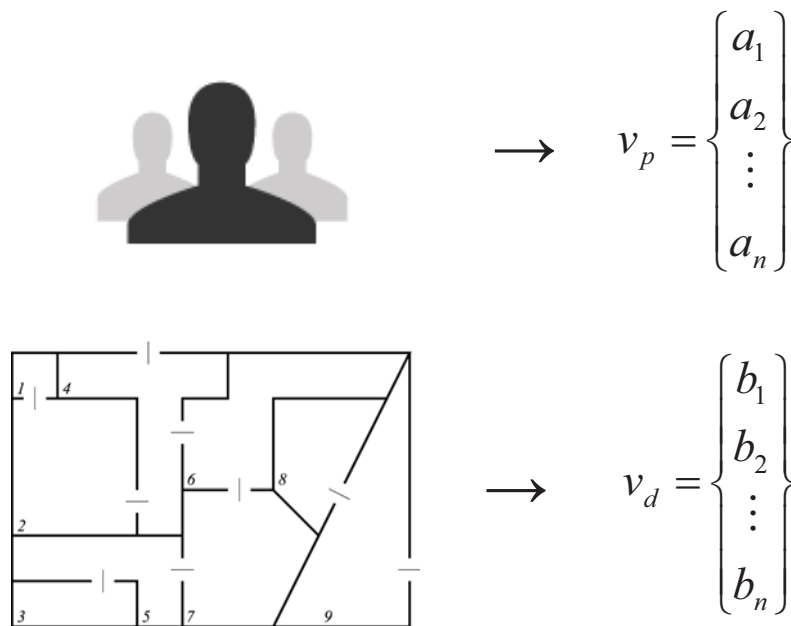


Abb. 7.1 Translation: Bildung von Vektoren

Jede ARCHITEKTUR ROUTINE kann für sich als eigenständige wissenschaftliche Hypothese betrachtet werden. Jede Operationalisierung eines Begriffs mit Hilfe der Methode der ALGORITHMISCHEN EXPLIKATION verkörpert die zum Zeitpunkt der Erstellung beste Annahme der jeweiligen Autorin oder des jeweiligen Autors dar, wie und durch welche Indikatoren die betreffenden Merkmale im digitalen Gebäudemodell *rational* erhoben werden kön-

<sup>4</sup> Barthes 1966, 191ff.

nen. Der Ausdruck „rational“ hat in diesem Zusammenhang nicht dieselbe Bedeutung wie „objektiv“ oder „beweisbar“. Bei expliziten Begriffsdefinitionen muss man sich damit begnügen, dass deren Vernünftigkeit wenigstens auf eingeschränkten Konsens stößt.<sup>5</sup> Bei der Abbildung realer Objekte oder Phänomene in eine maschinell verarbeitbare Repräsentation können semantische Lücken kaum vermieden werden. In keiner formalen Sprache lassen sich alle Zusammenhänge auswerten, die in einer so mächtigen Sprache wie der natürlichen ausgedrückt werden können. Die in dieser Arbeit beispielhaft entwickelten ARCHITEKTUR ROUTINE(n) sind als unverbindliche Vorschläge zur maschinellen Erfassung der betreffenden Gebäudemerkmale aufzufassen. Die damit verbundenen Begriffsdefinitionen werden niemandem aufgezwungen. Es handelt sich um freie Vereinbarungen vergleichbar mit einem Vertrag. Es bleibt jedem Vertragspartner selbst überlassen, diese zu akzeptieren oder nicht. Eine besonders wichtige Rolle spielt in diesem Zusammenhang die nachvollziehbare Darstellung der jeweils vom Entdeckungsalgorithmus angewendeten Operationen. Diese Transparenz wird durch die schriftliche Dokumentation der ALGORITHMISCHEN EXPLIKATION gewährleistet.

Der angefangene Merkmalskatalog im Anhang dieser Arbeit soll durch eine exemplarische, möglichst breit gestreute Mischung mehr oder weniger invarianter Operationalisierungen von Merkmalen dazu anregen, mit der in dieser Arbeit vorgestellten Methode zu experimentieren, andere, bessere oder vollständigere Operationalisierungen bereits vorgestellter Merkmale zu entwickeln, oder ARCHITEKTUR ROUTINE(n) zur Erhebung neuer Merkmale zu bilden. Ein Begriff lässt häufig mehrere unterschiedliche korrekte Operationalisierungen zu. Auf diese Weise kann und soll der Katalog beziehungsweise der Wortschatz wachsen. Mit jeder ARCHITEKTUR ROUTINE findet ein Begriff, welcher in der Alltagssprache ein bestimmtes Gebäudemerkmale beschreibt, Eingang in eine explizite Terminologie, die Menschen und Computern gleichsam zugänglich ist. Durch die wiederholte Anwendung des Verfahrens auf möglichst viele Begriffe, welche zur Beschreibung von Entwurfsaufgaben herangezogen werden, entsteht ein Vokabular, welches nicht nur an intersubjektive Vorstellungen geknüpft ist, sondern auch einen Dialog mit der Maschine erlaubt. Dabei sind selbst Routinen mit einer relativ niederen objektiven Gültigkeit (Invarianz) nicht wertlos. Sie können dazu dienen, das Denken über Entwurfsqualitäten anzuregen oder zu provozieren.<sup>6</sup>

Nach Kant kann das Wesen der ARCHITEKTUR ROUTINE(n) als transzendental bezeichnet, und jeder dieser Entdeckungsalgorithmen als die Implementierung eines synthetischen Urteils a priori interpretiert werden. Durch die ARCHITEKTUR ROUTINE(n) werden den Objekten im Gebäudedatenmodell Merkmalsausprägungen in Form von Zahlen oder Kategorien zugeteilt. Bei nominalskalierten Merkmalen erfolgt die Zuordnung nach dem Zweiwertigkeitsprinzip. Die Grundrissfigur eines Raums ist ein „Dreieck“ oder nicht, ein Raum ist „wohnlich“ oder nicht, Raum A direkter Nachbar von Raum B oder nicht. Im Falle ordinal- oder kardinalskalierten Merkmale erfolgt die Zuordnung zu den Merkmalskategorien graduell im Sinne eines Fuzzy Sets. Die Qualität der Orientierung eines Wohnraums ist „hoch“, „durchschnittlich“ oder „nieder“ beziehungsweise die Entfernung zweier Räume beträgt 5, 10 oder 100 Meter. Da Daten immer nur empirischer Natur sein können, ist auch das Er-

---

<sup>5</sup> Vgl. Eisenführ 2010, 5.

<sup>6</sup> Dies gilt insbesondere für die Betrachtung sehr subjektiver Begriffe wie Schönheit, Gemütlichkeit, Urbanität etc. In der Literatur findet sich eine Reihe kontroversieller Ansätze, wie man sich beispielsweise Themen der Ästhetik in der Architektur logisch-mathematisch nähern kann. Vgl. Kiemle 1967.



gebnis dieser Datenverarbeitung, also die konkret erfasste Merkmalsausprägung, in jedem Fall a posteriori. Die Verarbeitung der Daten als solche erfolgt jedoch nach Gesetzen a priori. Die dafür festgelegten Regelschritte (Operatoren) wurden bereits im Vorhinein, vor jeder Aufnahme von Daten im Zuge der *operationalen Definition* festgelegt. Das Erkenntnisermöglichende der Maschine ist der Entdeckungsalgorithmus, ein Programm, das die „Spontanität der Begriffe“<sup>7</sup> imitiert.<sup>8</sup>

*»Unsere Erkenntnis entspringt aus zwei Grundquellen des Gemüths, deren die erste ist, die Vorstellung zu empfangen (die Receptivität der Eindrücke), die zweite das Vermögen, durch diese Vorstellungen einen Gegenstand zu erkennen (Spontanität der Begriffe); durch die erste wird uns ein Gegenstand gegeben, durch die zweite wird dieser im Verhältnis auf jene Vorstellung [...] gedacht.«<sup>9</sup>*

## Ein neues Instrumentarium für die Architektur

Wie das Softwaretool ARCHILL.ES zeigt, eröffnen sich durch den Einsatz der ARCHITEKTUR ROUTINE(n) und die maschinelle Evaluierung von Entwurfslösungen Möglichkeiten für die Entwicklung einer gänzlich neuen Generation architekturenspezifischer, kollaborativer Softwaretools, welche die Entwurfstätigkeit von Architektinnen und Architekten in der Praxis wesentlich unterstützen können. Durch ein entwurfsbegleitendes maschinelles Feedback können die Sinne der Entwerferin und des Entwerfers erweitert werden oder eine rationale Entscheidungshilfe bei der Auswahl einer Variante geboten werden. Zugleich kann die spektrale Analyse von Entwurfslösungen für die Kommunikation architektonischer Konzepte starke und anschauliche Argumente liefern und auf einer rationalen Ebene einen wesentlichen Beitrag bei der Überzeugungsarbeit und Vermittlungstätigkeit von Architektinnen und Architekten zur Durchsetzung von baukünstlerischen Ideen leisten.

Für eine umfassende Beschreibung oder Beurteilung von Architektur ist die Betrachtung vielfältiger Aspekte notwendig. Die Beschränkung auf nur eine Methode oder einen thematisch eingegrenzten Satz von Methoden wäre dem Facettenreichtum architektonischer Aufgabenstellungen nicht angemessen. Einen ähnlich offenen Umgang pflegte bereits Christopher Alexander bei der Auflistung seiner 253 Entwurfsmuster indem er zu einer Erweiterung der *Pattern Language* durch individuelle Muster einlud:

*»Nun ergänze die Liste durch Hinzufügen deines eigenen Materials. Dinge, die in das Projekt einbezogen werden sollen, aber nicht als entsprechende Muster vorzufinden waren, schreib an passender Stelle in die Liste, zu anderen Mustern, die etwa die gleiche Größe und Wichtigkeit haben. [...]*

*Und selbstverständlich: wenn du irgendwelche Muster ändern willst, ändere sie, oft hat man eine persönliche Auffassung von einem Muster, die richtiger*

<sup>7</sup> Kant [erstveröffentlicht 1781] 2011, 107.

<sup>8</sup> Dotzler 2006, 45.

<sup>9</sup> Kant 2011, 107.

*oder für den Fall relevanter ist. In diesem Fall "beherrscht" man die Sprache am besten und eignet sie sich am wirksamsten an, wenn man die Änderungen an den entsprechenden Stellen des Buches einschreibt. Noch konkreter wird es sein, wenn man auch den Titel des Musters ändert, sodass es die eigenen Änderungen ausdrückt.«<sup>10</sup>*

Dass bei der Evaluierung von Gebäuden oder räumlichen Gegebenheiten der Fokus auf topologische Zusammenhänge nicht ausreicht, darauf hat bereits der italienische Architekt und Leiter des *Senseable City Lab* am MIT Carlo Ratti in seinen kritischen Betrachtungen der Methoden von Space Syntax hingewiesen.<sup>11</sup> Dasselbe gilt auch umgekehrt für ausschließlich geometrische Analysen bezüglich Distanzen, Größe, Form, Proportion etc. Bei der Evaluierung von Entwurfslösungen müssen topologische und geometrische Phänomene parallel betrachtet werden.

Das Vokabular der ARCHITEKTUR ROUTINE(n) ist erweiterbar und steht, neben der Integration bekannter Parameter wie Integration, Depth etc., der Modellierung vielfältiger Beurteilungs- und Analysekriterien offen. Durch Interfaces wie ARCHILL.ES können unterschiedlichste Betrachtungsweisen und Kriterien in Form von ARCHITEKTUR ROUTINE(n) operativ in den Entwurfsprozess eingegliedert werden. Als universelles Werkzeug für Architektinnen und Architekten beschränkt sich der Nutzen einzelner Methoden so nicht mehr auf praxisferne, avantgardistische Experimente. Durch die offene Konzeption der ARCHITEKTUR ROUTINE(n) ist alles integrierbar und willkommen: Space Syntax, Design Patterns, Bauordnung, Modulor, OIB, Feng Shui etc. - Wie es die Autoren der Tausend Plateaus Gilles Deleuze und Félix Guattari ausgedrückt haben: „Seid Vielheiten“ und „Laßt keinen General in euch aufkommen“.<sup>12</sup>

*»Der Informationsfluss wird von einem Graphen geregelt, der sozusagen das Gegenteil des hierarchischen Graphen ist ... Es gibt keinen Grund dass dieser Graph ein Baum sein muss. (einen solchen Graph nennen wir Karte). Vom Standpunkt [...] einer Guerillalogik aus findet man die Lösung ohne General für eine nicht zentrierte Vielheit, die eine endliche Anzahl von Zuständen und Signalen entsprechender Geschwindigkeiten enthält.«<sup>13</sup>*

Die Software ARCHILL.ES stellt bereits ein Basisrepertoire implementierter Merkmale zur Spezifikation von Gebäuden zur Verfügung. Dieses soll laufend erweitert werden. Als Umschlagplatz der Merkmalsmodule wurde die Onlineplattform WWW.ARCHILL.ES eingerichtet. Neu entwickelte Module können hier wie in einem *App Store* heruntergeladen und getestet werden. Es ist geplant, den Benutzerinnen und Benutzern von ARCHILL.ES in Zukunft auch die Implementierung eigener Module zu ermöglichen, welche dann als Plug-ins in das System geladen werden können. Auf diese Weise kann sich das Vokabular rascher erwei-

<sup>10</sup> Alexander [erstveröffentlicht 1977] 1995, XL.

<sup>11</sup> Vgl. Ratti 2003.

<sup>12</sup> Deleuze/Guattari, 1977, 28.

<sup>13</sup> Ebd.

tern, und funktionale Anforderungen an Gebäude können dadurch immer differenzierter spezifiziert werden. Expertinnen und Experten unterschiedlicher Fachrichtungen können ihr Know-how automatisiert in den Entwurfsprozess einfließen lassen. Die Bildung von ARCHITEKTUR ROUTINE(n) regt das allgemeine Nachdenken über Qualitäten eines architektonischen Entwurfs an. Dadurch kann die Anwendung auch im Bereich der Lehre sehr experimentellen und spielerischen Charakter annehmen.

## Faites vos jeux!

Weder für die Architektur noch für die Stadtplanung existieren strikte Regeln. Die Konsequenzen bestimmter Planungsentscheidungen können selten präzise vorhergesagt werden. In beide Fachrichtungen bedient man sich häufig diverser Kunstgriffe. Lösungen, die in einer spezifischen Situation adäquat sind, funktionieren jedoch nicht notwendigerweise in jeder Situation. Es gibt bislang keine Methode, wie das Funktionieren architektonischer Entwurfslösungen, bezogen auf den jeweiligen Kontext, objektiv überprüft werden könnte. Yona Friedman hat in seinem Buch *Toward a Scientific Architecture* bereits im Jahre 1975 darauf hingewiesen, dass es unter anderem deshalb unmöglich ist, architektonisches Entwerfen zu lehren. Die einzige Möglichkeit der Aneignung der für diese Tätigkeit notwendigen, häufig sehr intuitiven Fähigkeiten<sup>14</sup> besteht nach Friedman in der Imitation, das heißt in der Nachahmung des Stils eines Meisters (Meisterklassenprinzip).

*»But there are other disciplines, for the most part called sciences, in which strict rules have been established, at least for the most important areas, rules which are valid in a well-defined set of instances. In these disciplines the entire system of rules is as generalized as possible. In these disciplines, schools operate according to an arsenal of rules, which are available for the public (in books or films, for instance). Anybody who reads and understands these rules can apply them himself, without having to imitate the masters; and once he understands them, he can communicate them to anyone else. They are stated in such a way that in any instance it is clear whether they apply or not. I call these the teachable disciplines. In these areas of study, the personality of the teacher is of no importance.«<sup>15</sup>*

Im Gegensatz zur Architektur bieten andere Disziplinen, welche auf eine klare Axiomatik oder auf Grundsätze zurückgreifen können, die vereinbart wurden, und nicht weiter hinterfragt werden müssen, viel präzisere Möglichkeiten der Evaluierung. Das Spektrum reicht dort bis hin zum Führen von Beweisen. Nur wenn ein striktes Regelsystem existiert, kann auch eine rationale Beurteilung der Lösungen zu einem gestellten Problem erfolgen. Diese kann dann in der Regel auch maschinell durchgeführt werden. Die Anwendung quantitativer

---

<sup>14</sup> Der Psychologe Daniel Kahneman hat in seinem Buch *Schnelles Denken, langsames Denken* ausführlich darauf hingewiesen, dass Intuition nichts mit Zauberei zu tun hat, sondern eine Form von routiniertem, durch wiederholte Tätigkeiten angeeignetem Wissen darstellt, welches in manchen Situationen zu Tage tritt. Vgl. Kahneman 2011.

<sup>15</sup> Friedman 1975, 12.

Methoden bei der Beurteilung von Gebäudeentwürfen ergibt, bei Berücksichtigung der in Kapitel 6.1 beschriebenen, in jedem Fall kritisch zu betrachtenden Eigenheiten formalisierter Verfahren, auch Potentiale für die Lehre bei der Vermittlung von entwerferischen Fähigkeiten zukünftiger Architektinnen und Architekten. Dies führt unter anderem zu neuen Möglichkeiten einer spielerischen Aneignung.

In Bezug auf klare Regeln sind sich Spiel und Wissenschaft in mancher Hinsicht sehr ähnlich. Genauso wie die Wiederholbarkeit eine der wichtigsten Voraussetzungen empirischer Forschung darstellt, zeichnet sich das Spiel im Gegensatz zum Alltag unter anderem durch seinen repetitiven Charakter aus. In einem Mathematik Quiz beispielsweise können Rechenbeispiele genauso wie Spielzüge auf einem Schachcomputer trainiert werden. Durch das klare Axiomensystem der Arithmetik oder die präzisen Spielregeln im Schach kann die Beurteilung der einzelnen Rechenschritte oder Spielzüge automatisch erfolgen. Die Entwicklung von *Serious Games*, das sind Spiele mit einem ernsthaften Hintergrund, könnte sich neben den Erfolgen in anderen Disziplinen auch in der Architekturlehre als nützlich erweisen.<sup>16</sup>

In gewisser Weise gleicht das Entwerfen im Programm ARCHILL.ES einem Spiel. Wie auf jedem Spielbrett herrscht auch hier eine „*eigene und unbedingte Ordnung*“.<sup>17</sup> Die Spielregeln sind klar definiert, sie werden über die Slider am Mischpult aufgestellt. Durch die Verwendung expliziter Begriffe bei der Beschreibung von Entwurfparametern werden klare Vereinbarungen getroffen. Die Maschine übernimmt in der Folge die Funktion eines Schiedsrichters. Färbt sich ein Raum rot, ist er zum Beispiel „schlecht orientiert“. Wer darüber noch diskutieren will, bricht die inneren Gesetze des Spiels. Während der „Spieler“ blauen Farbflächen und High-Scores nacheifert, entsteht Schritt für Schritt ernsthafte Architektur.<sup>18</sup> So wie dem Konstruktivismus die Architektur als „Mutter aller Künste“ galt, hätte der Entwurf von Gebäuden nach dem *Digital Turn* vielleicht nun die Chance auch noch zum „nobelsten aller Spiele“ zu werden, und der Architekt, der *homo faber* würde so zum *homo ludens*.<sup>19</sup>

»Die nobelsten Spiele wie Schach bilden eine Kombinatorik von Orten in einem reinen Spatium, das unendlich viel tiefer ist als das reale Ausmaß des Schachbretts und die imaginäre Ausdehnung jeder Figur.«<sup>20</sup>

Eine solche Entwicklung bedeutet sicherlich nicht die Degradierung der Baukunst zu einem Spiel für Laien. Durch ein begleitendes, maschinelles Feedback könnten jedoch entwerferische Fähigkeiten spielerisch trainiert werden. Die unmittelbare Visualisierung der Erfolge und Misserfolge bei jeder Veränderung einer Entwurfslösung kann die Intuition für die nächste richtige Handlung schärfen. Entwerfen könnte in Zukunft, in Ergänzung zu den universitären Lehrangeboten, in Programmen wie ARCHILL.ES spielerisch und autodidakt erlernt beziehungsweise geübt werden.

<sup>16</sup> Vgl. Asendorpf 2011, 39.

<sup>17</sup> Vgl. Huizinga 2011, 19.

<sup>18</sup> Vgl. Emmerer 2014.

<sup>19</sup> Vgl. Huizinga 2011, 1.

<sup>20</sup> Deleuze [erstveröffentlicht 1969] 1992, 19.

## Fitness, Generative Methoden und das Andere

Da es sich bei dem Tool ARCHILL.ES gleichzeitig um ein Werkzeug zur Modellierung differenzierter Fitnessfunktionen von Gebäudeentwürfen handelt, eröffnet dies ebenso neue Möglichkeiten für die Anwendung generativer Entwurfsstrategien. Es wäre leicht möglich den DESIGN GOAL MANAGER und die Entdeckungsalgorithmen der ARCHITEKTUR ROUTINE(n) mit einem Formengenerator zu verknüpfen, welcher in der Lage ist, eine große Anzahl von Entwurfsvarianten automatisch zu generieren. Durch das in dieser Arbeit vorgestellte Verfahren der maschinellen Evaluierung von Entwurfslösungen, kann eine Reihung der Varianten und die Selektion der fittesten Lösungen maschinell erfolgen. Bei wiederholter Anwendung unter Einbezug geeigneter Mutations- und Rekombinations-Operationen eines evolutionären Algorithmus, könnten die emergenten Auswirkungen jeder Veränderung der Gebäudespezifikation unmittelbar, sozusagen in *Realtime*, mitverfolgt werden. Auf diese Weise wurden bereits von der Schweizer Forschungsgruppe *Kaisersrot*<sup>21</sup> interessante Ergebnisse erzielt. Durch den modular erweiterbaren expliziten Wortschatz der ARCHITEKTUR ROUTINE(n) könnten nach diesem Prinzip weiterführende Experimente erfolgen und facettenreiche, maschinell evaluierbare Gebäudespezifikationen zu hoch optimierten Entwurfslösungen führen. Der ultimative Turing-Test einer solchen abstrakten Entwurfsmaschine kann schlussendlich nur in der Teilnahme an anonymen Wettbewerben bestehen.

Der zentrale Aspekt bei der Evaluierung von Gebäudeentwürfen ist der Begriff der *Ähnlichkeit*. Die maschinelle Evaluierung einer Entwurfslösung besteht im Vergleich zweier Vektoren. Je ähnlicher  $v_p$  und  $v_d$  sind, desto eher erfüllt eine Entwurfslösung D auch das Programm P. Mathematisch kann die Ähnlichkeit zweier multidimensionaler Vektoren durch deren *Euklidischen Abstand* beschrieben werden. Fitness lässt sich in der Folge vereinfacht ausgedrückt als Ähnlichkeit einer Entwurfslösung zu einer fiktiven, optimalen Variante oder Gebäudespezifikation beschreiben.

$$v_p = \begin{Bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{Bmatrix} \leftrightarrow v_d = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix} \quad \|a - b\| = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

Ob nach diesem Prinzip jedoch ganz ohne menschliches Zutun etwas „radikal Neues“ entstehen kann, ist fraglich. Wirkliche Innovationen, architektonische Meisterleistungen entstanden ohne Zweifel nicht aufgrund des Befolgens irgendwelcher Vor-Schriften - Im Gegenteil: Das Andere, Neue, Kontroversielle oder das noch nicht Dagewesene hat in vielen Fällen gerade mit lateralem Denken (Querdenken), also dem Verlassen eines Programms zu tun. Genau an diesem Punkt spiegeln sich möglicherweise auch die Grenzen informierter, in

<sup>21</sup> [www.kaisersrot.ch](http://www.kaisersrot.ch)

sich abgeschlossener, das heißt überschaubarer Modelle; und auch von Analysemethoden, die sich auf die Anwendung der linearen Algebra beschränken. Um konventionelle Denkmuster zu durchbrechen ist es in kreativen Entwurfsprozessen mitunter nötig, nichtlinear zu denken, beispielweise bewußt nach der unwahrscheinlichsten Lösung zu suchen, vorliegende Informationen subjektiv zu bewerten oder selektiv zu verwenden, Assoziationen und Gedankensprünge zuzulassen, konventionelle Denkmuster in Frage zu stellen oder die in Raumprogrammen formulierten Rahmenbedingungen nicht als unveränderbar hinzunehmen.<sup>22</sup> Solche Begabungen menschlicher Entwerferinnen oder Entwerfer sind maschinell schwer imitierbar. Dies haben die teilweise ernüchternden Erkenntnisse der Erforschung von Künstlicher Intelligenz und Komplexität in den letzten fünf Jahrzehnten gezeigt. Zugleich wurde dadurch aber auch die Aufmerksamkeit der Wissenschaft für die verblüffenden, technisch bis heute nicht nachahmbaren, intuitiven Fähigkeiten des Menschen, unter anderem im Umgang mit Raum, geschärft.

Der architektonische Formfindungsprozess kann zwar, wie beispielsweise die Experimente von *Kaisersrot* gezeigt haben, weitestgehend automatisiert werden, wird jedoch auch dabei nicht gänzlich von einem entwerfenden Schöpfersubjekt befreit. Bei genauer Betrachtung kommt es im parametrischen Design im Grunde nur zu einer Transformation des Entwurfsprozesses. Parameter sind nichts anderes als, für einen spezifischen Fall subjektiv festgesetzte Variablen. Auch bei den ARCHITEKTUR ROUTINE(n) handelt es sich um gekapselte, formalisierte Subjektivität. Dieser Erkenntnis zufolge stellt die Entwicklung kollaborativer Methoden und offener, individuell adaptierbarer Systeme, welche die jeweiligen Fähigkeiten und Potenziale von Mensch und Maschine durch geeignete Interfaces symbiotisch verbinden, einen vielversprechenden Ansatz für die Einbindung von Computern in den architektonischen Entwurfsprozess dar.

Möglicherweise kann man bei der Entwicklung solcher Tools auch von den Strategien sozialer Netzwerke oder dem Page Ranking von Google etwas lernen. Das Problem des Suchenden ist hier nicht mehr das, unter Kreativen viel zitierte „weiße Blatt“, sondern vielmehr der Umgang mit dem nicht Überschaubaren. Man beginnt bei der Suche im Internet niemals bei Null. Im Gegenteil: Man startet mit einer infiniten, also unüberschaubaren Zahl von Möglichkeiten. Um in den grenzenlosen Datenbeständen des World Wide Web ein individuelles Suchziel zu erreichen oder brauchbare Antworten auf konkrete Fragen zu erhalten muß in einem iterativen Dialog mit der Maschine wiederholt definiert werden, was das Gesuchte „nicht ist“. Auch hierbei kann von einem *Modell* gesprochen werden: Die Abbildung der Welt basiert im Internet unter anderem auf der Indexierung von Inhalten und auf Data-Mining Algorithmen, welche die indexierten Dokumente und deren Verknüpfungen gewichten beziehungsweise bewerten und daraufhin den Suchenden individuelle Index-Karten zeichnen. Ein leistungsfähiges Werkzeug des Data-Minings stellt heute das bereits im Jahre 1982 von dem finnischen Ingenieur Teuvo Kohonen entwickelte Funktionsprinzip der sogenannten *Self Organizing Maps* (SOM) dar. Ludger Hovestad, Professor für CAAD an der ETH Zürich hat in dem Buch *EigenArchitektur* schon darauf hingewiesen, dass SOMs auf Grund ihrer besonderen Eigenschaften eine Klasse von Algorithmen darstellen, die möglicherweise dazu in der Lage wären, uns auch im Bereich der Architektur, bei dem Abenteu-

---

<sup>22</sup> Vgl. de Bono 1971.

er in nicht-informierten Lösungsräumen Neues zu entdecken, zu begleiten, und auf diese Weise heute die baukünstlerische Meisterschaft zu kultivieren.<sup>23</sup>

Wenn es nun schon einmal möglich ist, den Grad der Erfüllung eines Raumprogramms beziehungsweise die Ähnlichkeit zweier Entwurfslösungen auf mathematischem Weg zu ermitteln, so muss es genauso möglich sein, mit Hilfe von ARCHITEKTUR ROUTINE(n) das Andere, also maximal unähnliche oder konträre Lösungen zu finden. Die Translationsleistung der ARCHITEKTUR ROUTINE(n), die Fähigkeit in Gebäudedatenmodellen zu lesen und die interessierenden Gebäudemerkmalen als normierte, multidimensionale Vektoren abzubilden, können auch architektonische Entwurfslösungen der Analyse durch Self Organizing Maps zugänglich gemacht werden; erst durch Entdeckungsalgorithmen werden Building Information Models, wie schon Textdokumente bei der Plagiats- oder Rechtschreibprüfung, oder Rastergrafiken zur Gesichtserkennung über die Eingabe von Zahlen und Begriffen „googlebar“.

*»Wir haben jetzt das Land des reinen Verstandes nicht allein durchreiset und jeden Theil davon sorgfältig in Augenschein genommen, sondern es auch durchmessen und jedem Dinge auf demselben seine Stelle bestimmt. Dieses Land aber ist eine Insel und durch die Natur selbst in unveränderliche Grenzen eingeschlossen. Es ist das Land der Wahrheit (ein reizender Name), umgeben von einem weiten und stürmischen Oceane, dem eigentlichen Sitze des Scheins, wo manche Nebelbank und manches bald wegschmelzende Eis neue Länder lügt und, indem es den auf Entdeckungen herumschwärmenden Seefahrer unaufhörlich mit leeren Hoffnungen täuscht, ihn in Abenteuer verflechtet, von denen er niemals ablassen und sie doch auch niemals zu Ende bringen kann. Ehe wir uns aber auf dieses Meer wagen, um es nach allen Breiten zu durchsuchen und gewiß zu werden, ob etwas in ihnen zu hoffen sei, so wird es nützlich sein, zuvor noch einen Blick auf die Karte des Landes zu werfen, das wir eben verlassen wollen, und erstlich zu fragen, ob wir mit dem, was es in sich enthält, nicht allenfalls zufrieden sein könnten, oder auch aus Noth zufrieden sein müssen, wenn es sonst überall keinen Boden giebt, auf dem wir uns anbauen und uns wider alle feindselige Ansprüche gesichert halten können.«<sup>24</sup>*

## Rien ne va plus?

Die vorliegende Arbeit soll nicht als radikale Weiterführung des Optimierungsparadigmas in der Architektur, das heißt des seit Jahrhunderten bestehenden Diktats des Raumprogramms über den Architekturentwurf und dessen Ausweitung durch die Nutzung der Potenziale des Computers missverstanden werden.

Schon der Architekt William Mitchell hat in *The Logic of Architecture* darauf hingewiesen, dass sich die Auffassung, es könne eine abstrakte Zusammenstellung von Prädikaten vor und unabhängig von architektonischen Werken formuliert werden, und dass die Aufgabe

<sup>23</sup> Vgl. Hovestadt 2013, 66.

<sup>24</sup> Kant [Erstausgabe 1781] 2011, 258.

des Designers nur darin bestünde, diese Prädikate so umfassend und effizient wie möglich zu erfüllen, etwas naiv darstellt.<sup>25</sup> Als Architekt und beeindruckt von den Meisterleistungen der eigenen Disziplin, welche zu großen Teilen auf kreativer Freiheit beruhen, teilt der Autor der ARCHITEKTUR ROUTINE(n) diesen Standpunkt vollinhaltlich. Gerade die Strategien des Data-Minings im Internet haben gezeigt, dass deren enorme Möglichkeiten bei der Suche nach Neuem zugleich auch Potenziale restriktiver Kontrolle in sich bergen. Wie bei jedem Werkzeug kommt es ganz allein darauf an, wozu und wie man es benutzt.

Der Einsatz neuer Technologien bedeutet keine Schwächung des Status des Architekten als autonomes Schöpfersubjekt seiner Werke. Durch die Einbindung formalisierter Verfahren in den architektonischen Entwurfsprozess geht eine Veränderung des Konzepts des Künstlers als Autor einher, welche in anderen Kunstrichtungen schon länger in Gange ist: Auch in der Musik ist es durch die zunehmende Technisierung bereits zu grundlegenden Veränderungen gekommen. Zahlreiche Musiker legten in der zweiten Hälfte des 20. Jahrhunderts ihre alten Instrumente beiseite und begannen mit Beatboxen, Samplern und Mischpulten zu experimentieren.<sup>26</sup> Der DJ verschwindet durch die Technik des Samplings gewissermaßen als Autor hinter seinem elektronischen Equipment und scheint durch seine referenzielle Praxis der Funktion der künstlerischen Autonomie jeden Respekt zu verweigern.<sup>27</sup> Damit nimmt die DJ-Kultur in der Musik einen ähnlich ambivalenten Platz zwischen der Dekonstruktion und der Bewahrung der Idee des Künstlers ein, wie die Pop Art zuvor in den Bildenden Künsten.<sup>28</sup> Dennoch wird der DJ heute als eigenständiger Künstler wahrgenommen - wenn auch auf einer erweiterten Ebene. Im Umgang mit jedem Instrument gibt es Amateure und Experten: „*It's exactly the same with the sampler. Either you use it as a toy or you become a virtuoso.*“<sup>29</sup>

Prinzipiell kann man sagen, dass der Architektur ein wenig mehr Kalkül nicht schaden würde. Und Routinen zu bilden ist grundsätzlich nicht verwerflich. Aus dem in dieser Arbeit vollzogenen Zirkelschluss einer durchgehende Formalisierung des gesamten Kreislaufs, von der verbalen Beschreibung von Architekturqualitäten, bis hin zur Erfassung der interessierenden Eigenschaften konkreter Entwurfslösungen in einem Gebäudedatenmodell, erwächst in jedem Fall eine gänzlich neue Potenzialität. Die durch explizite Begriffe formulierten Vereinbarungen können durch die Implementierung in ARCHITEKTUR ROUTINE(n) in unterschiedlichen Anwendungsfeldern eingesetzt werden. Bei der Beschreibung von Gebäuden wird durch sie eine Verbindlichkeit zwischen Menschen und Maschinen geschaffen, welche bis heute durch die Vagheit der Alltagssprache nicht gegeben ist.

Die Wichtigkeit des Charakters einer „freien Vereinbarung“ kann in diesem Zusammenhang nicht oft genug erwähnt werden. Der Hinweis auf die in Kapitel 6.1 beschriebenen Grenzen oder Einschränkungen, welche (bei unreflektierter Anwendung) mit einer maschinellen Evaluierung von Entwurfslösungen einhergehen können, sollten auf keinen Fall unbeachtet bleiben. Die Potenziale und positiven Effekte der Einbindung von ARCHITEKTUR ROUTINE(n) in den architektonischen Entwurfsprozess können sich erst durch das aktive und kritische Bedienen dieses Regelwerks (bzw. der einzelnen Regler) durch Architektinnen

---

<sup>25</sup> Vgl. Mitchell 1996, 199 f.

<sup>26</sup> Vgl. Emmerer 2014.

<sup>27</sup> Vgl. Poschardt 1995, 375f.

<sup>28</sup> Ebd. 16.

<sup>29</sup> Ebd.



oder Architekten entfalten. Schon in der Einleitung dieser Arbeit wurde darauf hingewiesen: Die Erfüllung eines Raumprogramms garantiert keineswegs in jedem Fall gute Architektur – was immer das auch bedeuten mag. Es gibt eben bessere und schlechtere Programme – was immer das wiederum bedeuten mag.

Architektinnen und Architekten waren bei der Schaffung ihrer Meisterwerke zu jeder Zeit auf die verfügbaren Medien und (Hand)Werkzeuge angewiesen. Der unbändige Drang nach neuen, zeitgemäßen Entwurfsmethoden und Ausdrucksformen, führte dabei auch immer wieder zur Entwicklung neuer Tools. Diese haben sich im Laufe der Zeit stark verändert, bis hin zu einem universellen Werkzeug in Form der abstrakten Maschine. Um die medialen Funktionen ganz präzise und individuell an die persönlichen Bedürfnisse von Architektinnen und Architekten anpassen zu können, war stets ein gewisses Maß an Kontrolle über die eingesetzten Technologien notwendig. Dieses konnte, und kann auch in Zukunft nur erreicht werden, indem man die Werkzeuge seiner Zeit zu einem gewissen Grad mitgestaltet. *„Es ist wichtiger denn je, dass wir auch die Art, wie wir entwerfen, entwerfen.“*<sup>30</sup> In diesem Sinne: Seid Haus DJs<sup>31</sup> und nehmt die Hilfe der Maschine an. Entwickelt Routinen und „spielt“ auf neuen Instrumenten!

*»A great building must begin with the unmeasurable, must go through measurable means when it is designed and in the end must be unmeasurable.«*<sup>32</sup>

Louis Kahn 1930

---

<sup>30</sup> Hirschberg 2014, 38.

<sup>31</sup> Vgl. Emmerer 2014, 67.

<sup>32</sup> Louis Kahn 1930, zit. n. Green 1961, 3.



## MERKMALSKATALOG



No	Caption	Entity	Scale	Type
1	<i>Net Area (m)</i>	Room	V	Nettofläche
2	<i>Group Net Area</i>	Group (Rooms)	V	Gruppenfläche netto
3	<i>Order of Magnitude</i>	Room	O	Größenordnung
4	<i>Orthogonality</i>	Room	N	Orthogonal oder nicht
5	<i>Floor Plan Figure</i>	Room	N	Grundrissfigur
6	<i>Overview</i>	Room	V	Einsichtigkeit, Überschaubarkeit
7	<i>Window Count</i>	Room	A	Anzahl der Fenster
8	<i>Door Count</i>	Room	A	Anzahl der Zugänge
9	<i>Window Area</i>	Room	A	Gesamte Fensterfläche
10	<i>Window Quotient</i>	Room	V	Gesamte Fensterfläche/ Nettofl.
11	<i>Quality of Orientation</i>	Room	O	Qualität der Orientierung (Licht)
12	<i>Morning Sun</i>	Room	N	Morgensonne, Ostorientierung
13	<i>Light from two Sides</i>	Room	N	Licht von mehr als einer Seite Pattern Language No159
14	<i>Escape Way Length (m)</i>	Room	V	Fluchtweglänge
15	<i>Integration</i>	Room	V	Space Syntax
16	<i>Walk Distance (m)</i>	Room - Room	V	kürzeste fußläufige Entfernung
17	<i>Direct Access (Doorway)</i>	Room - Room	N	Zugänglichkeit (Türverbindung)
18	<i>Adjacency</i>	Room - Room	N	Nachbarschaft
19	<i>Same Level</i>	Room - Room	N	Im selben Geschoß
.				
.				
.				
n				

V ... Verhältnisskala

A ... Absolutskala

N ... Nomialskala

O ... Ordinalskala



**Net Area****№ 1**

Room Property



Autor:	Martin Emmerer	Vokabular:
Datum:	17.02.2012	-
Skala:	Verhältnisskala	
Einheit:	m <sup>2</sup> - square meter	
Wertemenge:	R {0, ..., n}	

**(a) Analytische Definition**

Das Merkmal AREA NET ist der Flächeninhalt der ebenen begehbaren Bodenfläche eines Raums, die zum Aufenthalt von Personen oder zur Möblierung benutzt werden kann. Die Größe eines Raums wird in Quadratmetern gemessen. Bereiche mit einer Bodenneigung von über 2%, wie zum Beispiel Rampen, sowie Ausnehmungen im Boden, wie sie beispielsweise im Bereich von Galerien oder Stiegenhäusern auftreten, zählen nicht zur Größe eines Raums. Da Bereiche mit einer Raumhöhe von weniger als 1,5m von den meisten Menschen in aufrechter Haltung nicht oder schlecht begehbare sind, werden diese der Größe eines Raumes nicht angerechnet. (Zum Beispiel im Bereich von Dach- oder Wandschrägen, Treppenläufe etc.)

**(b) Operationale Definition**

Die Nettogröße eines Raums erhält man durch Berechnung des Flächeninhalts des Polygons NETPOLY der betreffenden Instanz von TRoom im Gebäudemodell abzüglich der Flächeninhalte aller Polygone des Arrays mit dem Namen *Holes*. Die Flächenberechnung kann mithilfe der Gaußschen Trapezformel erfolgen.

$$A = \frac{1}{2} \cdot \left| \sum_{i=1}^n (y_i + y_{i+1}) \cdot (x_i - x_{i+1}) \right|$$

**(c) Implementierung**

```
function DISCOVER(ModelPTR:TEntities; RoomNr:integer):Real;
var
  i:integer;
begin
  Result:=Poly2DArea(ModelPTR^.RL[RoomNr].Netpoly);
  for i:=0 to high(ModelPTR^.RL[RoomNr].Holes) do
    Result:=Result - Poly2DArea(ModelPTR^.RL[RoomNr].Holes[i]);
end;
```

## Group Net Area

№ 2

Group Property



★★★★☆☆

Autor:	Martin Emmerer	Vokabular:
Datum:	17.10.2013	
Skala:	Ratio Scale	-
Einheit:	m - meter	
Wertemenge:	R {0, ...,n}	

### (a) Analytische Definition

Das Merkmal GROUP NET AREA ist die Summe der Nettoflächen aller Räume einer namentlich definierten Raumgruppe.

### (b) Operationale Definition

Das Merkmal GROUP AREA NET wird durch Erhebung der Nettoflächen aller Räume innerhalb einer Gruppe mit Hilfe der ARCHITEKTUR ROUTINE No.1 - NET AREA festgestellt. Ein Raum gehört zu einer Gruppe G, wenn die betreffende Instanz des Datenobjekts ROOM im Parameter GROUP den String G enthält.

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; Group:string):real;
begin
  Result:=0;
  With ModelPTR^ do
    For i:=0 to high(RL) do
      if not (RL[i].Deleted) and (RL[i].SpaceType=Room) then
        if equalstrings(Arguments, RL[i].Group) then
          Result:= Result + DISCOVER_NetArea(ModelPTR, i);
    end;

  INCLUDED ROUTINES -----

  DISCOVER_NetArea(ModelPTR, i);
```

Delphi Code

## Order of Magnitude 60:80

№ 3

Room Property



★★★★☆

Autor:	Martin Emmerer	Vokabular:
Datum:	17.02.2012	
Skala:	Ordinalskala	(0) small
Einheit:	-	(1) medium
Wertemenge:	N {0, 1, 2}	(2) large

### (a) Analytische Definition

Das Merkmal ORDER OF MAGNITUDE bildet Kategorien von Räumen unterschiedlicher Größe. Dabei wird zwischen den Größenordnungen bis 60m<sup>2</sup> (SMALL), 60-80m<sup>2</sup> (MEDIUM) und >80m<sup>2</sup> (LARGE) unterschieden.

### (b) Operationale Definition

Durch die ARCHITEKTUR ROUTINE *Order of Magnitude* erfolgt eine Abbildung der Messergebnisse der verhältnisskalierten ARCHITEKTUR ROUTINE No.1 - NET AREA auf eine Ordinalskala. Nach der Ermittlung der exakten Nettogröße erfolgt die auf den Schwellenwerten 60m<sup>2</sup> und 80m<sup>2</sup> basierende Zuweisung der ganzzahligen Resultate 0, 1 und 2, welche für die Kategorien „Small“, „Medium“ und „Large“ stehen.

### (c) Implementierung

```
function DISCOVER(Room):Integer
var A:Real;
begin
  A:=NetArea(Rom);
  if A<60 then Result:=0
    else if A>80 then Result:=2
      else Result:=1;
end;
```

Delphi Code

## Orthogonality

№ 4

Room Property



★★★★★

Autor:	Martin Emmerer	Vokabular:
Datum:	14.09.2013	
Skala:	Nominalskala	(0) not orthogonal
Einheit:	-	(1) orthogonal
Wertemenge:	$N \{0,1\}$	

### (a) Analytische Definition

Das Merkmal ORTHOGONAL erfasst, ob der Grundriss eines Raums ausschließlich orthogonale Ecken hat oder nicht. Zwei Kanten mit einem Innenwinkel von  $0.0^\circ$  oder  $360.0^\circ$  gelten nicht als orthogonale Ecken.

### (b) Operationale Definition

Die Orthogonalität eines polygonalen Raums wird durch das Messen des inneren Winkels an jeder Ecke des Polygons  $R.NetPoly$  festgestellt. Wenn nur einer der Winkel weder  $90.0^\circ$  noch  $180.0^\circ$  noch  $270.0^\circ$  aufweist, erfolgt der Rückgabewert „not orthogonal“ (0). Entsprechen alle Winkel einem dieser Kriterien, so ist das Result „orthogonal“ (1).

Verfeinerungspotential: Es könnte ergänzend eine Toleranz von beispielsweise  $\pm 1.0^\circ$  vorgesehen werden.

### (c) Implementierung

```
function Orthogonal(R:Room):Integer;
begin
  Count:=0;
  L:=length(R.NetPoly);
  for i:=0 to L-1 do
  begin
    Ang:= GetVertexangle(R.NetPoly,i);
    if (Ang<>90) and (Ang<>180) and (Ang<>270) then inc(Count);
  end;
  if Count>0 then Result:=0 else Result:=1;
end;
```

Delphi Code



## Floor Plan Figure

№ 5

Room Property



★★★★☆☆

Autor:	Martin Emmerer	Vokabular:
Datum:	14.01.2013	(0) Triangle
Skala:	Nominalskala	(1) Rectangle
Einheit:	-	(2) Square
Wertemenge:	$N \{0,1,2,3,4,5,6,7\}$	(3) L-Shape
		(4) T-Shape
		(5) Cross
		(6) Star
		(7) Other

### (a) Analytische Definition

Das Merkmal FLOOR PLAN FIGURE beschreibt die Form des Grundrisses eines einfachen polygonalen Raums. Die Wertemenge beschränkt sich auf die sieben Kategorien Triangle (0), Rectangle (1), Square (2), L-Shape (3), T-Shape (4), Cross (5), Star (6). Zusätzlich wird die Kategorie Other (7) für alle anderen möglichen Figuren eingeführt. Da der Begriff Square eine Sonderform eines Rechtecks darstellt, und somit jedes Quadrat auch ein Rechteck ist, erfüllen diese beiden Merkmalsausprägungen genaugenommen nicht das Exklusivitätskriterium von Nominalskalen, da sich die Kriterien nicht gegenseitig ausschließen. Obwohl dies von der mathematischen Definition abweicht, soll festgesetzt werden, dass ein Quadrat eine eigenständige Kategorie darstellt.

### (b) Operationale Definition

Die Grundrissfigur eines Raums im Building Information Model kann durch die Analyse der Polygons POLY der betreffenden Instanz des Datenobjekt ROOM erfolgen. Um entscheiden zu können, in welche Kategorie die Figur des Polygons fällt, müssen in allgemeiner Form Merkmale definiert werden, durch welche sich die unterschiedlichen Figuren geometrisch voneinander eindeutig unterscheiden. Zu diesem Zweck werden folgende Festlegungen gemacht, welche teilweise aus den mathematischen Definitionen der betreffenden Figuren stammen bzw. auf Beobachtungen beruhen, welche im Bezug auf die meisten einfachen Polygone zufriedenstellende Entscheidungsgrundlage darstellen:

0-7) Konvexe Kanten mit einem Innenwinkel von mehr als  $150^\circ$  und konkave Kanten mit einem Innenwinkel von weniger als  $210^\circ$  werden nicht betrachtet.

0) Hat ein Polygon 3 Kanten, die nicht auf einem Punkt zusammenfallen oder in einer Linie liegen, ist seine Figur ein DREIECK.

- 1) Hat ein Polygon 4 Kanten, dessen Innenwinkel jeweils  $90^\circ$  (+/- einer Toleranz von  $0.5^\circ$ ) betragen und eine Länge von mehr als 0m aufweisen, dann ist seine Figur ein RECHTECK.
- 2) Haben die erste und dritte bzw. die zweite und vierte Kante eines Rechtecks dieselbe Länge, dann handelt es sich bei der Figur um ein QUADRAT.
- 3) Hat ein Polygon fünf konvexe und eine konkave Ecken, dann ist seine Figur ein L-SHAPE.
- 4) Hat ein Polygon sechs konvexe und zwei konkave Ecken, dann ist seine Figur ein T-SHAPE.
- 5) Hat ein Polygon acht konvexe und vier konkave Ecken, dann ist seine Figur ein CROSS.
- 6) Hat ein Polygon mehr als 4 Kanten und gleich viele konkave wie konvexe Ecken, so ist seine Figur ein STAR.
- 7) Die Figuren von Polygonen mit anderen Eigenschaften fallen unter die Kategorie OTHER.

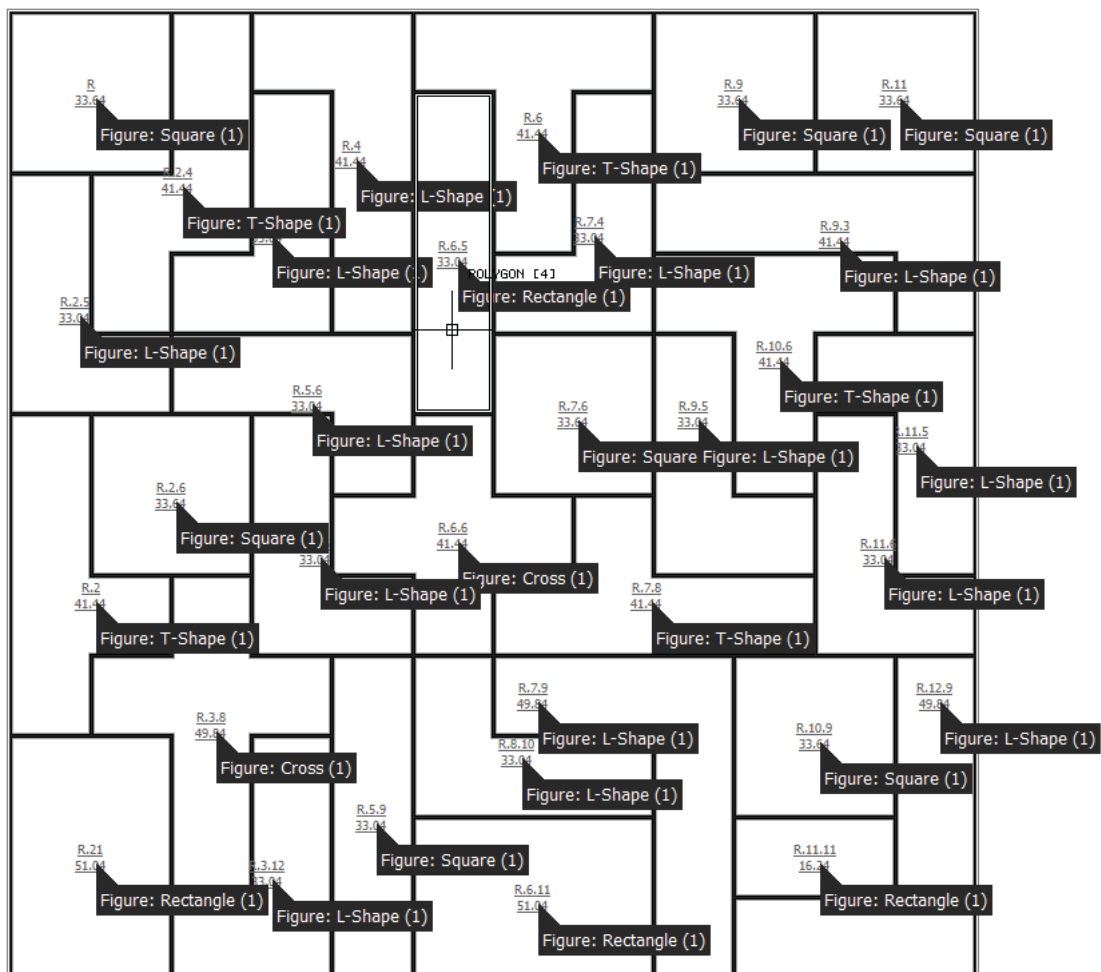


Abb.1 Testanwendung 32 Räume

## (c) Implementierung

```

function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):integer;
begin
  Poly2Figure(ModelPTR^.RL[RoomNr].NetPoly, 0.5)
end;

INCLUDED ROUTINES -----

function Poly2Figure(P:T2DPoly; OrthoTolerance:real):integer;
var MajorEdges,concave,convex,i,L:integer;
    isOrtho:boolean;
    E:array[0..3] of Double;
    a:real;
begin
  L:=length(P);
  if L=3 then begin Result:=0; exit; end;

  Result:=7;
  MakeClockwise(P);
  isOrtho:=true;
  concave:=0;
  convex:=0;
  Majoredges:=0;

  for i:=0 to high(P) do
  begin
    A:=InnerVertexAngle(P,i);
    if (Abs(A-90) > OrthoTolerance)
    and (Abs(A-270)> OrthoTolerance)
    and (Abs(A-180)> OrthoTolerance) then isOrtho:=false;

    if Abs(A-180)>30 then
    begin
      inc(Majoredges);
      if A>180 then inc(convex);
      if A<180 then inc(concave);
    end;
  end;

  Case Majoredges of
    4:if isOrtho then
      begin
        E[0]:=Point2DDist(P[0],P[1]);
        E[1]:=Point2DDist(P[1],P[2]);
        E[2]:=Point2DDist(P[2],P[3]);
        E[3]:=Point2DDist(P[3],P[0]);
        if Equal(E[0],E[2],7) and Equal(E[1],E[3],7) then
          if Equal(E[0],E[1],7) then Result:=2 else Result:=1;
        end;
      6 :if (concave=5) and (convex=1) then Result:=3;
      8 :if (concave=6) and (convex=2) then Result:=4;
     12:if (concave=8) and (convex=4) then Result:=5;
    end;
    if (L>4) and (convex=concave) then Result:=6;
  end;
end;

```

## Overview

№ 6

Room Property



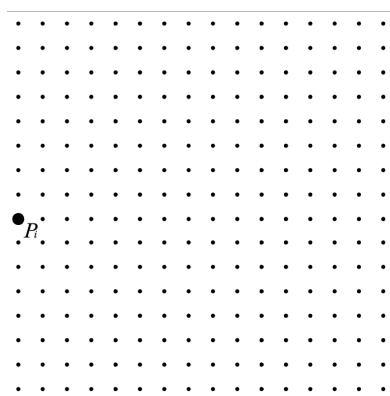
Autor:	Martin Emmerer	Vokabular:
Datum:	17.7.2012	
Skala:	Verhältnisskala	-
Einheit:	% - percent	
Wertemenge:	$R \{0, \dots, n\}$	

### (a) Analytische Definition

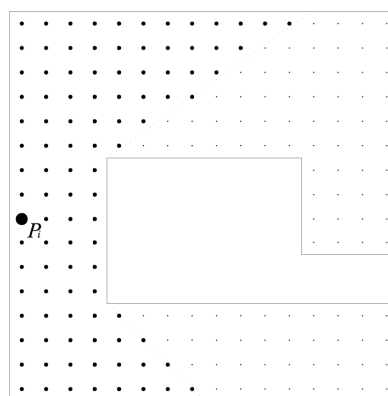
Ein Raum wird als übersichtlich bezeichnet wenn von möglichst vielen Standpunkten in dessen Inneren eine Vielzahl von Bereichen des gesamten Raumes überblickt werden kann. Unübersichtliche Räume hingegen haben tote Winkel, das sind von bestimmten Standpunkten aus betrachtet nicht-einsichtige Bereiche.

### (b) Operationale Definition

Das Merkmal OVERVIEW (Übersichtlichkeit) eines Raums kann festgestellt werden, indem für eine endliche Anzahl von Standpunkten ermittelt wird, wie viel Prozent der anderen Standpunkte jeweils von diesem aus eingesehen werden kann. Diese Prozentzahl soll als *Übersichtsfaktor* bezeichnet werden. Die Standpunkte sollten möglichst gleichmäßig verteilt und die Anzahl der Raumgröße angemessen sein. Die Wahl der Standpunkte kann durch eine orthogonale Rasterung des betreffenden Raums erfolgen. Dabei hat ein Kompromiss zwischen aufzuwendender Rechenzeit und der Genauigkeit bei der Erfassung geometrisch kleinteiliger Raumsituationen zu erfolgen. In Anbetracht üblicher Grundrisskonfigurationen wird das Achsmaß des Rasters auf 0.5m festgesetzt. Der in Prozent erfasste Gesamtwert der Übersichtlichkeit eines Raums erfolgt durch die Bildung des arithmetischen Mittels aller Übersichtsfaktoren.



$n=256$   $P_i=100\%$   
 $s_i=256$



$n=200$   $P_i=50\%$   
 $s_i=100$

Abb.1 Beispiel Übersichtsfaktor Standpunkt P

Der Algorithmus arbeitet folgendermassen: Als erstes wird das kleinste rechteckige Feld ermittelt, in das der betrachtete Raum eingeschrieben werden kann. (Xmin, Xmax, Ymin, Ymax). Danach wird dieses Feld gleichmäßig gerastert und eine Liste jener Punkte (PList) erstellt, die innerhalb des Polygons liegen. Hierzu dient die Programmfunktion PointinPoly2D. Im letzten Schritt wird für jeden innenliegenden Punkte festgestellt, wieviele andere Punkte von diesem ausgehend eingesehen werden können. Dazu wird das Polygon Netpoly jeweils mit dem Verbindungssegment des Punktepaars geschnitten. Existiert zumindest ein Schnittpunkt, dann kann der betreffende Punkt im Raum auch nicht eingesehen werden. Die Summe aller eingesehenen Punkte durch die maximal einsehbaren Punkte (konvexes Polygon) ergibt den Wert für die Übersichtlichkeit des Raums in Prozent.

### (c) Implementierung

```
function DISCOVER(R:Room):real;
begin
  GetOverview(R.NetPoly, Result)
end;

INCLUDED ROUTINES -----

function GetOverview(Poly:T2Dpoly; Var Overview:real):boolean;
var
  Count,MAX,i,ii,L,NUM:integer;
  XMin,XMax,YMin,YMax,
  dx,dy,
  X,Y:real;
  XP,PT:T2DPoint;
  PList:T2DPointlist;
  convex:boolean;

begin
  Result:=true;
  Convex:=true;
  for i:=0 to high(Poly) do
    if GetPoly2DinnerVertexAngle(Poly,i)>180 then begin convex:=false; break; end;
    if convex then begin Overview:=1.0; exit; end;

    //GET LIMITS
    XMin:= Double_Infinity; XMax:=-Double_infinity;
    YMin:= Double_Infinity; YMax:=-Double_infinity;
    For i:=0 to high(Poly) do
      begin
        if Poly[i,0]<XMin then XMin:=Poly[i,0];
        if Poly[i,0]>XMax then XMax:=Poly[i,0];
        if Poly[i,1]<YMin then YMin:=Poly[i,1];
        if Poly[i,1]>YMax then YMax:=Poly[i,1];
      end;
    if (XMin<Double_Infinity) and (XMax>-Double_infinity)
    and (YMin<Double_Infinity) and (YMax>-Double_infinity) then
      begin
        Result:=true;

        //GET POINTS
```

```
setlength(PList,0);
if Result then
begin
  Y:=YMin;
  While Y<YMax do
  begin
    Y:=Y+0.5;
    X:=XMin;
    While X<XMax do
    begin
      X:=X+0.5;
      PT:=Point2d(X,Y);
      if PointinPoly2D(PT, Poly) then
      begin
        L:=length(PList);
        Setlength(PList,L+1);
        PList[L]:=PT;
      end;
    end;
  end;
end;

//GET OVERVIEW
NUM:=0; MAX:=0;
For i:=0 to high(PList) do
  for ii:=0 to high(PList)-i do if (i<>ii) then
  begin
    inc(MAX);
    if not FastPolyXSeg2D(Poly,TRUE,Seg2D(PList[i],PList[ii]), FALSE, XP)
    then inc(NUM);
  end;
Overview:=NUM / MAX;
end;

end;
```

Delphi Code

## Window Count

№ 7

Room Property



★★★★★

Autor:	Martin Emmerer	Vokabular:
Datum:	02.03.2013	
Skala:	Absolutskala	-
Einheit:	Pcs - pieces	
Wertemenge:	N {0,1,2, ...,n}	

### (a) Analytische Definition

Das Merkmal WINDOW COUNT misst die Anzahl der Fenster eines Raumes. Ein Fenster gilt dann als zu einem Raum zugehörig, wenn es in einem Wandabschnitt sitzt, welcher den betreffenden Raum vom Außenraum trennt und somit durch das Öffnen des Fensters eine direkte Verbindung zum Außenraum hergestellt werden kann.

### (b) Operationale Definition

Die Anzahl der Fenster (Window Count) kann durch Zählung jener Instanzen des Datenobjekts TOpening im Array OL des betreffenden TRoom-Objects erfolgen, deren Property Objecttype den Wert Window hat und deren Property Parent1 oder Parent2 dem Index des entsprechenden Raums entspricht.

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):integer;
var
  i,ii:integer;
begin
  Result:=0;
  for i:=0 to high(ModelPTR^.ZL) do
    if not(ModelPTR^.ZL[i].Deleted) then
      if ModelPTR^.ZL[i].Objecttype=Window then
        for ii:=0 to high(ModelPTR^.ZL[i].Linkedrooms) do
          if (ModelPTR^.ZL[i].Linkedrooms[ii]=RoomNr)then Inc(Result);
        end;
      end;
  end;
```

Delphi Code

## Door Count

№ 8

Room Property



★★★★★

---

Autor:	Martin Emmerer	Vokabular:
Datum:	02.03.2013	
Skala:	Absolutskala	-
Einheit:	pcs - pieces	
Wertemenge:	N {0,1,2, ...,n}	

---

### (a) Analytische Definition

Das Merkmal DOOR COUNT misst die Anzahl der Zugänge (Türen) eines Raumes.

### (b) Operationale Definition

Die Anzahl der Türen (Door Count) kann durch Zählung jener Instanzen des Datenobjekts TOpening im Array OL des betreffenden TRoom-Objects erfolgen, deren Property Objecttype den Wert Door hat und deren Property Parent1 oder Parent2 dem Index des entsprechenden Raums entspricht.

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):integer;
var
  i,ii:integer;
begin
  Result:=0;
  for i:=0 to high(ModelPTR^.ZL) do
    if not(ModelPTR^.ZL[i].Deleted) then
      if ModelPTR^.ZL[i].Objecttype=Door then
        for ii:=0 to high(ModelPTR^.ZL[i].Linkedrooms) do
          if (ModelPTR^.ZL[i].Linkedrooms[ii]=RoomNr)then Inc(Result);
        end;
      end;
  end;
```

Delphi Code



## Window Area

№ 9

Room Property



★★★★☆

---

Autor:	Martin Emmerer	Vokabular:
Datum:	27.02.2012	-
Skala:	Verhältnisskala	
Einheit:	m <sup>2</sup> - square meter	
Wertemenge:	R {0, ..., n}	

---

### (a) Analytische Definition

Das Merkmal WINDOW AREA misst die Summe aller Fensterflächen in einem Raum.

### (b) Operationale Definition

Die Gesamtfensterfläche eines Raums kann durch das Summieren sämtlicher Fensterflächen eines Raums im Gebäudedatenmodell erhoben werden. Die Zugehörigkeit eines Fensters zu einem Raum besteht dann, wenn dessen Indexwert im Parameterarray LINKED-ROOMS enthalten ist. Die Fläche jedes Fensters wird durch das Produkt seiner Breite (Parameter WIDTH) und Höhe (Parameter HEIGHT) berechnet.

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):real;
var
  A:real;
  i:integer;
begin
  Result:=0;
  for i:=0 to high(ModelPTR^.ZL) do
    if ModelPTR^.ZL[i].Objecttype=Window then
      if (ModelPTR^.ZL[i].Linkedrooms[0]=RoomNr)
      or (ModelPTR^.ZL[i].Linkedrooms[1]=RoomNr)then
        begin
          A:=ModelPTR^.ZL[i].Width * ModelPTR^.ZL[i].Height;
          Result:=Result + A;
        end;
  end;
end;
```

Delphi Code

## Window Quotient

№ 10

Room Property



★★★★☆☆

---

Autor:	Martin Emmerer	Vokabular:
Datum:	17.10.2013	-
Skala:	Verhältnisskala	
Einheit:	% - percent	
Wertemenge:	R {0, ...,1}	

---

### (a) Analytische Definition

Der WINDOW QUOTIENT beschreibt das Verhältnis der Fensterfläche eines Raums zu dessen Nettfläche. Dieses Merkmal ist beispielsweise im Wohnbau oder für Gastronomiebetriebe von Bedeutung. In manchen nationalen Bautechnikverordnungen oder Wohnbauförderungsgesetzen wird für dieses Maß eine Mindestanforderungen wie 10% vorgegeben (REF 1).

### (b) Operationale Definition

Der Fensterquotient wird durch die Division des Ergebnisses der bereits definierten Funktion WINDOW AREA durch das Ergebnis der Funktion NET AREA gebildet.

### (c) Implementierung

```
function DISCOVER(R:Room):real;
begin
  Result:= DISCOVER_WindowArea(R) / DISCOVER_NetArea(R);
end;
```

INCLUDED ROUTINES -----

```
DISCOVER_GetWindowArea(Room):real;
DISCOVER_GetNetarea(Room):real;
```

Delphi Code

### (d) Referenzen

REF 1: Vgl. OIB Richtlinie 3, Hygiene, Gesundheit und Umweltschutz: Kapitel 9, Belichtung und Beleuchtung, Österreichisches Institut für Bautechnik, Oktober 2011, S.6ff.

## Quality of Orientation

№ 11

Room Property



Autor:	Martin Emmerer	Vokabular:
Datum:	14.09.2013	(0) none
Skala:	Ordinalskala	(1) low
Einheit:	-	(2) medium
Wertemenge:	N {0,1,2,3}	(3) high

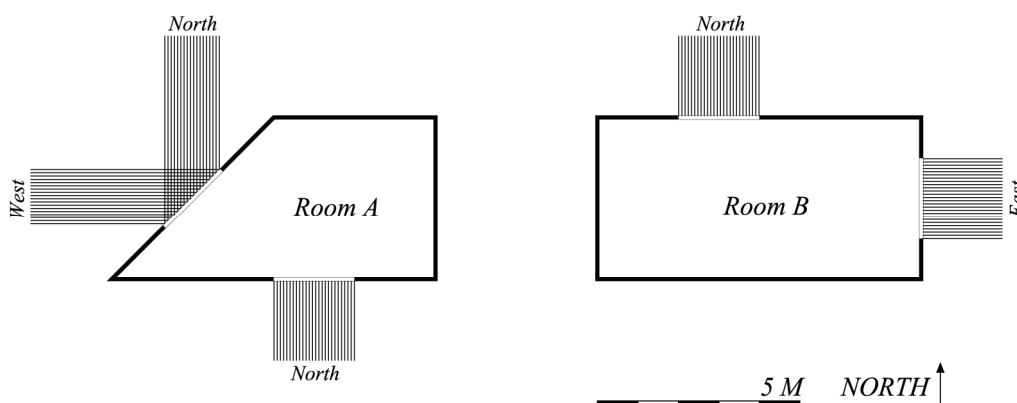
### (a) Analytische Definition

Das Merkmal QUALITY OF ORIENTATION beschreibt in sehr allgemeiner und vereinfachter Form die Qualität der Orientierung eines Raumes (z.B. Wohnraum) in Bezug auf die geographische Ausrichtung und Größe der Fensteröffnungen. Die Qualität der Orientierung eines Raums gilt als „hoch“, wenn mindestens eines der Fenster anteilig nach Süden oder Westen ausgerichtet ist. Fensteröffnungen mit einer Fläche kleiner als 0.5m<sup>2</sup> werden dabei als nicht relevant betrachtet.

Die Qualität der Orientierung von Räumen ohne Süd- oder West-Orientierung, welche aber zumindest nach Osten orientierte Fenster aufweisen, wird als „mittelmäßig“ eingestuft. Die Qualität von Räumen mit ausschließlich nordorientierten Fenstern wird als „nieder“ bezeichnet. Räume ohne Fenster erhalten das Prädikat „keine“ Qualität der Orientierung.

### (b) Operationale Definition

Die Qualität der Orientierung eines Raums kann festgestellt werden, indem alle Fenster in die vier Himmelsrichtungen Norden, Süden, Osten und Westen projiziert werden. Abbildung 1 zeigt diesen Vorgang anhand von zwei beispielhaften Räumen. Durch die prozentuelle Zerlegung der Gesamtfensterfläche in die vier Himmelsrichtungen kann der Flächenanteil pro Himmelsrichtung berechnet werden.



Zerlegung der Fensterflächen

## (c) Implementierung

```

function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):integer;
var
  A,W,A_Sum, A_West, A_East, A_South, A_North :real;
  VList:TValueList;      //ORDER: North/West/South/East
  i:integer;
begin
  Result:=0; A_West:=0; A_East:=0; A_South:=0; A_North:=0; A_Sum :=0;
  for i:=0 to high(ModelPTR^.ZL) do
    if not(ModelPTR^.ZL[i].Deleted) then
      if ModelPTR^.ZL[i].Objecttype=Window then
        if length(ModelPTR^.ZL[i].Linkedrooms)=2 then
          if (ModelPTR^.ZL[i].Linkedrooms[0]<0)
          or (ModelPTR^.ZL[i].Linkedrooms[1]<0) then
            if (ModelPTR^.ZL[i].Linkedrooms[0]=RoomNr)
            or (ModelPTR^.ZL[i].Linkedrooms[1]=RoomNr) then
              begin
                GetBorderSegOrientation(
                  ModelPTR^.ZL[i].BLInst.Segments[0],
                  Seg2D(ModelPTR^.ZL[0].Zeropoint, ModelPTR^.ZL[0].YAxis),
                  ModelPTR^.ZL[i].Height, A, VList);
                A_Sum :=A_Sum + A;
                A_North:=A_North + VList[0];
                A_West :=A_West + VList[1];
                A_South:=A_South + VList[2];
                A_East :=A_East + VList[3];
              end;
            if A_Sum>0 then
              begin
                if (A_North>0.5) then Result:=1;
                if (A_East>0.5) then Result:=2;
                if (A_West>0.5) or (A_South>0.5) then Result:=3;
              end;
            end;
          end;
        end;
      end;
    end;
  end;

INCLUDED ROUTINES -----
procedure GetBorderSegOrientation(SEG, VDIRECT:T2DSegment; H:real;
                                var Area:real;
                                var VALUES:TValueList);
var
  i:integer;
  D_ANG, A, W:real;
  SEG1:T2DSegment;
begin
  Setlength(VALUES,4);
  Area:=Seg2dlength(SEG)*H;
  for i:=0 to 3 do
    begin
      VALUES[i]:=0;
      SEG1:=SEG2DROTATE(VDIRECT, VDIRECT[0], i*90 - 90);
      D_ANG:=ResSeg2DAngle(SEG1[0],SEG1[1],SEG[0],SEG[1]);
      if (D_ANG>270) or (D_ANG<90) then
        begin
          //CALCULATE PERCENTAGE
          if D_ANG>270 then A:=360-D_ANG else A:=D_ANG;
          VALUES[i]:= (90-A)/90 * Area;
        end;
      end;
    end;
  end;
end;

```

## Morning Sun

**№ 12**

Room Property



Autor:	Martin Emmerer	Vokabular:
Datum:	02.03.2013	
Skala:	Nominal	(0) No
Einheit:	-	(1) Yes
Wertemenge:	$N \{0,1\}$	

### (a) Analytische Definition

Das Merkmal *Morning Sun* beschreibt, ob in einen Raum am Morgen Sonnenlicht fällt. Damit am Morgen in einen Raum Sonnenlicht fällt, muss dieser zumindest anteilig nach Osten orientiert sein. Dies bedeutet, dass der Raum ein Fenster besitzen muss, das zumindest teilweise nach Osten gerichtet ist und eine gewisse Größe haben muss.

Verfeinerungspotential: Beschattung durch benachbarte Gebäude, Bäume, Topographie etc., Berücksichtigung des tatsächlichen horizontalen und vertikalen Sonnenwinkels (Azimuth und Altitude) unter Berücksichtigung des wahren Längen- und Breitengrades des Grundstücks.

### (b) Operationale Definition

Zur Feststellung des Merkmals *Morning Sun* eines Raums wird zuerst der Ostanteil jedes Fensters berechnet. Dazu werden alle Fensteröffnungen in Richtung Osten projiziert. Abbildung 1 zeigt diesen Vorgang anhand von zwei beispielhaften Räumen. Ist die Summe der anteilig nach Osten orientierten Projektionsflächen der Fenster größer als  $0,5\text{m}^2$ , dann wird das Resultat „Yes“ (1) zurückgegeben; Kleinere Ostflächenanteile bewirken den Rückgabewert „No“ (0),

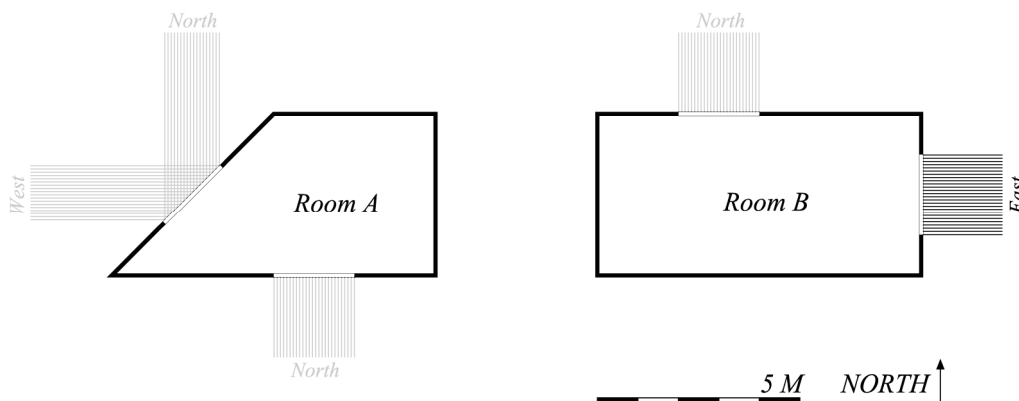


Abb.1 Zerlegung der Fensterflächen

## (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):integer;
var
  A_North, A_West, A_South, A_East, A_Sum:real;
begin
  Get_WindowAreas(ModelPTR, RoomNr, A_North, A_West, A_South, A_East, A_Sum);
  if A_East>0.5 then Answer:=1 else Answer:=0;
end;
```

INCLUDED ROUTINES -----

```
procedure Get_WindowAreas(ModelPTR:TEntitiesPTR; RoomNr:integer;
                          var A_North, A_West, A_South, A_East, A_Sum:real);
```

```
var
  A:real;
  i:integer;
  VList:TValuelist; //North/West/South/East
```

```
begin
  A_North:=0; A_West:=0; A_South:=0; A_East:=0; A_Sum :=0;
```

```
  for i:=0 to high(ModelPTR^.ZL) do
    if not(ModelPTR^.ZL[i].Deleted) then
      if ModelPTR^.ZL[i].Objecttype=Window then
        if length(ModelPTR^.ZL[i].Linkedrooms)=2 then
          if (ModelPTR^.ZL[i].Linkedrooms[0]=RoomNr)
            or (ModelPTR^.ZL[i].Linkedrooms[1]=RoomNr) then
            begin
              GetBorderSegOrientation(ModelPTR^.ZL[i].BLInst.Segments[0],
                                      Seg2D(ModelPTR^.ZL[0].Zeropoint,
                                              ModelPTR^.ZL[0].YAxis),
                                      ModelPTR^.ZL[i].Height A, VList);

              A_Sum :=A_Sum + A;
              A_North:=A_North + VList[0];
              A_West :=A_West + VList[1];
              A_South:=A_South + VList[2];
              A_East :=A_East + VList[3];
            end;
```

```
end;
```

INCLUDED ROUTINES -----

```
GetBorderSegOrientation: s. ARCHITEKTUR ROUTINR No 6
```

## Light from Two Sides

№ 13

Room Property



Autor:	Martin Emmerer	Vokabular:
Datum:	14.09.2014	
Skala:	Nominalskala	(0) one-sided
Einheit:	-	(1) two-sided
Wertemenge:	$N \{0, 1\}$	

### (a) Analytische Definition

Laut dem Entwurfsmuster Nr.159 der *Pattern Language* von Christopher Alexander „Licht von zwei Seiten in jedem Raum“, weisen Räume, in die Licht von zwei Seiten einfallen kann eine besondere räumliche Qualität auf, welche einseitigem Lichteinfall in jedem Raum vorzuziehen sind.

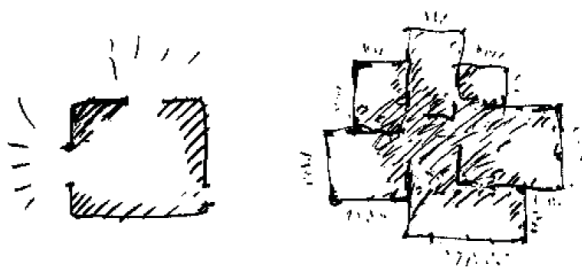


Abb.1 Skizze Christopher Alexander

*Wenn Menschen die Wahl haben, halten sie sich lieber in Räumen auf, die Licht von zwei Seiten haben, während Räume, die nur von einer Seite Licht haben, unbenutzt und leer bleiben. Dieses Muster entscheidet vielleicht mehr als irgendein anderes Muster über Gelingen' oder Mißlingen eines Raums. Die Anordnung des Tageslichts in einem Raum und das Vorhandensein von Fenstern auf zwei Seiten sind etwas Wesentliches. Wenn man ein Zimmer mit Licht von nur einer Seite baut, kann man ziemlich sicher sein, daß das reine Geldverschwendung ist. Die Leute werden das Zimmer, wenn es sich vermeiden läßt, nicht betreten. Natürlich - wenn alle Räume nur Licht von einer Seite haben, müssen die Leute sie benutzen. Aber wir können annehmen, daß sie sich dort im Grunde nicht wohlfühlen, daß sie am liebsten gar nicht dort wären und gern wieder gehen würden - weil wir natürlich genau wissen, was die Leute tun, wenn sie die Wahl haben. [...] Leg jeden Raum so an, daß er zumindest auf zwei Seiten an den Außenraum grenzt, und dann setz in diese Außenwände Fenster, sodaß in jeden Raum natürliches Licht aus mehr als einer Richtung fällt. (REF 1)*

## (b) Operationale Definition

Ob ein Raum von zwei Seiten belichtet wird, kann festgestellt werden, indem zunächst die Anzahl der Fenster gezählt wird. Hat der betreffende Raum mehr als ein Fenster, muss festgestellt werden, ob zwischen den Achsen mindestens zwei der Fenster ein Winkel von weniger als  $90^\circ$  oder mehr als  $270^\circ$  besteht. Ist das der Fall, dann gilt das Kriterium „Licht von zwei Seiten“ als erfüllt.

## (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomPos:integer):integer;
var
  Ang:real;
  i,ii, C:integer;
  SEG1, Windowaxis, Northaxis:T2DSegment;
  Direction:array [0..3] of boolean;

function getcategory(Ang:real):integer;
// 0=North:45-135, 1=West :135-225, 2=South:225-315, 3=East :0-45 or 315-360
begin
  Result:=0;
  if (Ang>45 ) and (Ang<=135) then Result:=0;
  if (Ang>135) and (Ang<=225) then Result:=1;
  if (Ang>225) and (Ang<=315) then Result:=2;
  if ((Ang>316) and (Ang<=360))
  or ((Ang>=0 ) and (Ang<=45 )) then Result:=3;
end;

begin
  for i:=0 to 3 do Direction[i]:=false;
  for i:=0 to high(ModelPTR^.ZL) do
    if not(ModelPTR^.ZL[i].Deleted) then
      if ModelPTR^.ZL[i].Objecttype=Window then
        for ii:=0 to high(ModelPTR^.ZL[i].Linkedrooms) do
          if (ModelPTR^.ZL[i].Linkedrooms[ii]=RoomPos)then
            if length(ModelPTR^.ZL[i].BLInst.Segments)>0 then
              begin
                SEG1:=ModelPTR^.ZL[i].BLInst.Segments[0];
                Windowaxis:=SEG2DROTATE(SEG1, SEG1[0], 90);
                Northaxis :=Seg2d(ModelPTR^.ZL[0].Zeropoint,
                                   ModelPTR^.ZL[0].Yaxis);
                ANG:=ResSeg2DAngle(Windowaxis[0],Windowaxis[1],
                                   Northaxis[0],Northaxis[1]);
                C:=getcategory(Ang);
                Case C of
                  0:Direction[0]:=true;
                  1:Direction[1]:=true;
                  2:Direction[2]:=true;
                  3:Direction[3]:=true;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
  Result:=0;
  for i:=0 to 3 do if Direction[i] then inc(Result);
end;
```



## (d) Referenzen

REF 1: Christopher Alexander, Eine Mustersprache. Städte Gebäude Konstruktion, Löcker Verlag, Wien 1995, S.811ff.

Der Originaltext ist für die analytische Definition stark gekürzt worden. Selbstverständlich war sich Alexander dessen bewusst, dass beispielsweise auch die Lage der Fenster in Bezug auf die geometrische Form des Raumes entscheidend ist. Die weiteren Aspekte wurden jedoch vorerst aus den Betrachtungen ausgeklammert.

## Escape Way Length

№ 14

Room Property



★★★★☆☆

Autor:	Martin Emmerer	Vokabular:
Datum:	14.01.2013	
Skala:	Verhältnisskala	-
Einheit:	M - meter	
Wertemenge:	R {0, ..., n}	

### (a) Analytische Definition

Das Merkmal ESCAPE WAY LENGTH misst die Länge des Fluchtweges aus einem bestimmten Raum. Die Fluchtweglänge ist für die Sicherheit eines Gebäudes im Brandfall entscheidend. Mit Fluchtweg ist der kürzeste (direkte) Weg ins Freie gemeint. Dabei wird von der ungünstigsten Stelle des betreffenden Raums ausgegangen (REF 1).

### (b) Operationale Definition

Die Fluchtweglänge kann ermittelt werden, indem der V-Graph (Visibility Graph) eines Gebäudes gebildet wird (vgl. Kapitel 4.2), und in diesem danach ausgehend von jeder Node vom Typ gnExit ein Dijkstra Algorithmus gestartet wird. Auf diese Weise kann ausgehend von jeder Fluchttüre der jeweils kürzeste Weg zur ungünstigsten Stelle des untersuchten Raums ermittelt werden. Der kürzeste von allen dieser (kürzesten, ungünstigsten) Wege wird als Fluchtweg betrachtet und dessen Länge als Resultat übergeben.

### (c) Implementierung

```
Function DISCOVER(ModelPTR:TEntitiesPTR; RoomPos:integer;
                 var DIST:real):boolean;
var
  LocalGraph:TGraph;
  i,ii, Pos, Endpos:integer;
  D, ExitWayLength:real;
  Found:boolean;

begin
  Result:=true;
  GRAPH_Clone(ModelPTR^.V_GRAPH, LocalGraph);    //MAKE A LOCAL COPY
  ExitWayLength:=999999;
  Pos:=-1;

  With Localgraph do
  for i:=0 to NodeCount-1 do
    if (NList[i].NTYPE=gnExit) then
      begin
```

```
GRAPH_Start_Dijkstra(Localgraph, i);
D:=0;
Found:=false;
for ii:=0 to NodeCount-1 do
  if (NList[ii].Index=RoomPos) then
    if (NList[ii].DISTFROMSTART>D) and (NList[ii].DISTFROMSTART<999999)
      then begin Found:=true; Endpos:=ii; D:=NList[ii].DISTFROMSTART;end;
  if Found and (D<ExitWayLength) then
    begin Pos:=Endpos; Exitwaylength:=D; end;
end;

Dist:=Exitwaylength;
GRAPH_Clear(Localgraph);
Freeandnil(Localgraph);
end;
```

Delphi Code

## (d) Referenzen

REF 1: Diese Definition wurde zur Implementierung etwas vereinfacht: Allgemein lässt sich sagen, dass der Fluchtweg eine gewisse vorgeschriebene Länge (z.B. 40m lt. steirischer Bauordnung) entweder zu einem anderen Brandabschnitt, zu einem Treppenhaus mit einem Ausgang ins Freie oder zu einem direkt ins Freie führenden Ausgang von jedem Punkt innerhalb jedes Aufenthaltsraumes nicht überschreiten darf. Je nach Gebäudetyp und regionaler Bauordnung variieren die vorgeschriebenen maximalen Fluchtweglängen, sowie das Erfordernis von nur einem oder zwei getrennten Fluchtwegen.

## Integration

№ 15

Room Property



Autor:	Martin Emmerer	Vokabular:
Datum:	17.10.2013	
Skala:	Verhältnisskala	-
Einheit:	% - percent	
Wertemenge:	R {0, ..., 1}	

### (a) Analytische Definition

Das Merkmal INTEGRATION stellt einen zentralen Aspekt bei den Raumanalysen von SPACESYNTAX dar und wurde von Bill Hillier und Julienne Hanson in „The Logic of Space“ präzise charakterisiert (REF 1):

*The measure of integration*

*The notion of depth has already been introduced, in the sense that axial or convex segments were either many steps – that is, deep – from buildings or from the carrier, or a few steps – that is shallow – from the carrier or the buildings. Relations of depth necessarily involve the notion of asymmetry, since spaces can only be deep from other spaces if it is necessary to pass through intervening spaces to arrive at them. The measure of relative asymmetry generalises this by comparing how deep or shallow it theoretically could be – the least depth existing when all spaces are directly connected to the original space, and the most when all spaces are arranged in a unilinear sequence away from the original space, i.e. every additional space in the system adds one more level of depths.*

*Relative asymmetry (or relative depth) can therefore be thought of more simply as the measure of integration. Of course, for all but the smallest systems these calculations should be done by computer. (...) A mean figure is the mean RA from all points in the system. This is the general measure of integration for the system as a whole.*

### (b) Operationale Definition

*To calculate relative asymmetry from any point, work out the mean depth of the system from the space by assigning a depth value to each space according to how many spaces it is away from the original space, summing these values and dividing by the number of spaces in the system less one (the original space). Then calculate relative asymmetry as follows:*

$$RA = \frac{2(MD - 1)}{k - 2}$$

*where MD is the mean depth and k the number of spaces in the system. This will give a value between 0 and 1, with low values indicating a space from which the system is shallow, that is a space which tends to integrate the system, and high values a space which tends to be segregated from the system.*

Hilliers Ausführungen beinhalten im Grunde schon die operationale Definition des Merkmals INTEGRATION. Die Berechnung dieses Parameters erfolgt im Gebäudedatenmodell über den Access-Graphen (AC\_Graph). Die Entfernung eines Raums zu jedem anderen Raum kann mit Hilfe des Algorithmus von Dijkstra erfolgen. Die Kanten des Graphen müssen dabei mit dem Wert 1.0 gewichtet sein, da das Maß der Tiefe (DEPTH) im Sinne von Spacesyntax die Anzahl der zu durchschreitenden Räume und nicht die wahre Weglänge abbildet.

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; RoomNr:integer):real;
var
  LocalGraph:TGraph;
  Startpos, Count,i:integer;
  MD:real;
begin
  Result:=1;
  GRAPH_Clone(ModelPTR^.AC_GRAPH, LocalGraph); //MAKE A LOCAL COPY

  startpos:=-1;
  for i:=0 to high(Localgraph.NList) do
    if RoomNr=Localgraph.NList[i].INDEX then startpos:=i;
  if startpos<0 then exit;
  GRAPH_Start_Dijkstra(Localgraph, Startpos);

  Count:=0; MD:=0;
  With Localgraph do
  for i:=0 to NodeCount-1 do
  if NList[i].INDEX<>RoomNr then
  if NList[i].DISTFROMSTART<999999 then
    begin
      inc(Count);
      MD:=MD+NList[i].DISTFROMSTART;
    end;

  if Count>0 then
  begin
    MD:=MD/Count;
    Result:=2*(MD-1) / (Count-1);
  end
  else Result:=1;

  Result:=1-Result;
  GRAPH_Clear(Localgraph);
  Freeandnil(Localgraph);
end;
```

Delphi Code

### (d) Referenzen

REF 1: Hillier Bill, Hanson Julienne, The Social Logic of Space, Cambridge University Press 1984, S.108f.

## Walk Distance

**№ 16**

Relation Property



★★☆☆☆

Autor:	Martin Emmerer	Vokabular:
Datum:	14.01.2013	
Skala:	Verhältnisskala	-
Einheit:	m - meter	
Wertemenge:	$R \{0, \dots, n\}$	

### (a) Analytische Definition

Das Merkmal WALK DISTANCE beschreibt die Länge des kürzesten Weges zwischen zwei Räumen. Die Grundrissfigur der Räume ist für die Ermittlung der Weglänge nicht von Bedeutung. Gemessen wird ausgehend von den jeweiligen Türöffnungen.

### (b) Operationale Definition

Die fußläufige Distanz zwischen zwei Räumen kann ermittelt werden, indem der V-Graph (Visibility Graph) eines Gebäudes gebildet wird (vgl. Kapitel 4.2) und in diesem danach mit Hilfe des Dijkstra Algorithmus die Entfernungen jeder Tür des einen Raums zu jeder Tür des anderen Raums ermittelt wird. Die Länge des kürzesten dieser Wege wird dann durch die Summation der Längen dessen Segmente berechnet.

### (c) Implementierung

```
function DISCOVER(var BMODEL:TEntities; R1,R2:integer; var DIST:real):boolean;
var
  LocalGraph:TGraph;
  D:real;
  StartPos, Endpos, i,ii:integer;
  Found:boolean;
begin
  Result:=false;
  Found:=false;

  D:=Double_Infinity;

  GRAPH_Clone(BMODEL.V_GRAPH, LocalGraph);    //MAKE A LOCAL COPY

  With Localgraph do
  for i:=0 to NodeCount-1 do
    if NList[i].NTYPE=gnDoor then
      if NList[i].Index=R1 then
        begin
          GRAPH_Start_Dijkstra(Localgraph, i);
          for ii:=0 to NodeCount-1 do
```

```
        if NList[ii].NTYPE=gnDoor then
          if NList[ii].Index=R2 then
            begin
              if (NList[ii].DISTFROMSTART<D)
                then begin
                  Found:=true;
                  Startpos:=i; Endpos:=ii;
                  D:=NList[ii].DISTFROMSTART;
                end;
            end;
          end;
        end;

    if found then
    begin
      Dist:=D;
      Result:=true;
    end;
  end;
```

Delphi Code

## Direct Access (Doorway)

№ 17

Relation Property



★★★★★

Autor:	Martin Emmerer	Vokabular:
Datum:	02.04.2014	
Skala:	Nominalskala	(0) not connected
Einheit:	-	(1) connected
Wertemenge:	N {0,1}	

### (a) Analytische Definition

Das Merkmal *Direct Access* beschreibt, ob zwei Räume A und B miteinander direkt verbunden sind. Eine direkte Verbindung liegt vor, wenn die betreffenden Räume durch eine Tür so miteinander verbunden sind, dass kein weiterer Raum betreten werden muss, um von Raum A in Raum B zu gelangen.

### (b) Operationale Definition

Ob zwei Räume direkt miteinander verbunden sind, lässt sich nachweisen, indem festgestellt wird, ob die betreffenden Räume im Building Information Model eine Türverbindung aufweisen. Existiert zumindest eine Instanz des Datenobjekts OPENING, dessen Parameterarray LINKEDROOM[0..1] den Indexwert von Raum A und Raum B enthält, dann sind die Räume durch eine Tür verbunden (connected).

### (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; R1,R2:integer):integer;
var
  i:integer;
begin
  Result:=0;
  for i:=0 to high(ModelPTR^.ZL) do
    With ModelPTR^.ZL[i] do if not ModelPTR^.ZL[i].Deleted then
      if ObjectType=Door then
        if length(linkedrooms)=2 then
          if ((linkedrooms[0]=R1) and (linkedrooms[1]=R2) )
            or ((linkedrooms[0]=R2) and (linkedrooms[1]=R1) )
            then begin Result:=1; exit; end;
end;
```

Delphi Code



# Adjacency

**№ 18**

Relation Property



★★★★★☆

Autor:	Martin Emmerer	Vokabular:
Datum:	10.03.2013	
Skala:	Nominalskala	(0) not adjacent
Einheit:	-	(1) adjacent
Wertemenge:	N {0,1}	

## (a) Analytische Definition

Das Merkmal Adjacency beschreibt, ob zwei Räume benachbart sind oder nicht. Eine direkte Nachbarschaft liegt vor, wenn zwei Räume aneinander grenzen, das heißt gemeinsame Außenwände sowie dasselbe Fußbodenniveau aufweisen. Für das Prädikat „benachbart“ ist keine Türverbindung notwendig, es muss nur theoretisch möglich sein, eine, direkte Verbindung herzustellen.

## (b) Operationale Definition

Die Nachbarschaft zweier Räume kann erhoben werden, indem festgestellt wird, ob entweder das Parameterarray ADJ\_R von Raum A den Indexwert von Raum B enthält, oder umgekehrt, das Parameterarray ADJ\_R von Raum B den Indexwert von Raum A enthält.

## (c) Implementierung

```
function DISCOVER(ModelPTR:TEntitiesPTR; R1,R2:integer):integer;
var i:integer;

begin
  Result:=0;
  if (R1>-1) and (R1<high(ModelPTR^.RL)) then
    if not ModelPTR^.RL[R1].Deleted then
      for i:=0 to high(ModelPTR^.RL[R1].ADJ_R) do
        if ModelPTR^.RL[R1].ADJ_R[i]=R2 then begin Result:=1; exit; end;
end;
```

Delphi Code

## Same Level

**Nº 19**

Relation Property



★★★★★

---

Autor:	Martin Emmerer	Vokabular:
Datum:	8.02.2013	
Skala:	Nominalskala	(0) same Level
Einheit:	-	(1) different Levels
Wertemenge:	N {0,1}	

---

### (a) Analytische Definition

Das Merkmal SAME LEVEL beschreibt, ob sich zwei Räume im selben Geschöß befinden oder nicht. Zwei Räume befinden sich im selben Geschöß, wenn diese dasselbe Fußbodenniveau haben.

### (b) Operationale Definition

Ob zwei Räume im Datenmodell dasselbe Fußbodenniveau aufweisen, kann durch das Vergleichen des Parameters Level von Raum A mit dem von Raum B festgestellt werden.

### (c) Implementierung

```
Function DISCOVER(ModelPTR:TEntitiesPTR; R1,R2:integer):integer;  
begin  
  if ModelPTR^.RL[R1].Level <> ModelPTR^.RL[R2].Level  
  then Result:=1  
  else Result:=0;  
end;
```

Delphi Code





# LITERATUR



## **AHO 2004**

Ausschuss der Verbände und Kammern der Ingenieure und Architekten für die Honorarordnung e.V.(Hg.): Neue Leistungsbilder zum Projektmanagement in der Bau- und Immobilienwirtschaft (=AHO Schriftenreihe 19, Köln 2004 )

## **Alexander 1964**

Alexander, Christopher: Notes on the Synthesis of Form, Harvard University Press, Cambridge-London 1964

## **Alexander 1995**

Alexander, Christopher: Eine Mustersprache. Städte Gebäude Konstruktion, Löcker Verlag, Wien 1995

## **Anderson 2013**

Anderson, Chris: Das Ende der Theorie. Die Datenschwemme macht Wissenschaftliche Methoden obsolet, in: Geiselberger, Heinrich; Moorstedt, Tobias (Hg.): Big Data. Das neue Versprechen der Allwissenheit, Suhrkamp Verlag, Berlin 2013, S.124-130

## **Arndt 2011**

Arndt, Hans Werner: Die Entwicklungsstufen von Leibniz' Begriff einer Lingua Universalis, [http://www.vordenker.de/downloads/arndt\\_leibniz-lingua-universalis.pdf](http://www.vordenker.de/downloads/arndt_leibniz-lingua-universalis.pdf), in: <http://www.vordenker.de> 20.10.2014  
[Erstveröffentlichung in: Hans-Georg Gadamer (Hg.): Das Problem der Sprache, Wilhelm Fink, München 1967, S.71-79]

## **Asendorpf 2011**

Asendorpf, Dirk: Die Nacht-Falter – Computerspiele helfen Forschern, komplexe Probleme zu lösen, in: Die Zeit 27.01.2011, 5

## **Barker-Plummer 2008**

Barker-Plummer, Dave et al: Tarski's World – Revised and Expanded, CSLI Publications, Stanford 2008

## **Barker 2009**

Barker, J. S. F.: Defining fitness in natural and domesticated populations, in: J. van der Werf, J. van der (Hg.): Adaptation and Fitness in Animal Populations. Springer-Verlag, Heidelberg 2009, S. 3-14

## **Barthes 1966**

Barthes, Roland: Die strukturalistische Tätigkeit, in: Kimmich, Dorothee (Hg.): Texte zur Literaturtheorie der Gegenwart, Reclam, Stuttgart 2008, S.214-222

## **Barwise/Etchemendy 2005**

Barwise, Jon; Etchemendy John: Sprache, Beweis und Logik – Aussagen- und Prädikatenlogik, Mentis Verlag, Paderborn 2005

**Bauer u.a. 1992**

Bauer, Friedrich L., Goos Gerhard: Informatik. Eine einführende Übersicht, Bd.2, Springer Verlag 1992

**Baumann 2006**

Baumann, Peter: Erkenntnistheorie, Verlag J. B. Metzler, Stuttgart 2006

**Betsch 2005**

Betsch, Tilmann: Wie beeinflussen Routinen das Entscheidungsverhalten, in: Psychologische Rundschau 56(2005) H.4, S. 261-270.

**Biggs 1998**

Biggs, N. L. et al.: Graph Theory 1736-1936, Clarendon Press, Oxford 1998 [Erstveröffentlichung 1977]

**Blanke 1985**

Blanke, Detlev: Internationale Plansprachen. Eine Einführung, Akademie Verlag, Berlin 1985

**Blanke 1996**

Blanke, Detlev: Leibniz und die Lingua Universalis, in: Sitzungsberichte der Leibniz-Sozietät 13 (1996) 5. S.27-35. Online unter: [http://leibnizsozietat.de/wp-content/uploads/2012/10/02\\_blanke.pdf](http://leibnizsozietat.de/wp-content/uploads/2012/10/02_blanke.pdf)

**Bono 1971**

Bono, Edward de: Laterales Denken: ein Kursus zur Erschließung ihrer Kreativitätsreserven, Rowohlt, Reinbek 1971

**Bortz 2006**

Bortz, Jürgen; Döring, Nicola: Forschungsmethoden und Evaluation, Springer Verlag, Heidelberg 2006

**Box/Draper 1987**

Box, George Edward Pelham; Draper, Norbert R.: Empirical Model Building and Response Surfaces, John Wiley & Sons, New York 1987.

**Bridgman 1927**

Bridgman, Percy Williams: The Logic of Modern Physics, MacMillan Edition, New York 1927

**Bridgman 1938**

Bridgman, Percy Williams: Operational Analysis, in: Philosophy of Science 5 (1938) 2, S.114-131

**Bridgman 1959**

Bridgman, Percy Williams: The Logic of Modern Physics' after Thirty Years, in: Daedalus 88 (1959) 3, S.518 - 526

**Carnap 1959**

Carnap, Rudolf: Induktive Logik und Wahrscheinlichkeit, Springer, Wien 1959

**Cramer/Kamps 2008**

Cramer, Erhard; Kamps, Udo: Grundlagen der Wahrscheinlichkeitsrechnung und Statistik. Ein Skript für Studierende der Informatik, der Ingenieur- und Wirtschaftswissenschaften, Springer, Berlin-Heidelberg 2008

**Dilthey 1990**

Dilthey, Wilhelm: Logik und System der Philosophischen Wissenschaften. Vorlesungen zur Erkenntnistheoretischen Logik und Methodologie (1864-1903), Gesammelte Schriften XX.Band, Vandenhoeck & Ruprecht, Göttingen 1990

**Deleuze 1992**

Deleuze, Gilles: Woran erkennt man den Strukturalismus, Merve Verlag, Berlin 1992

**Deleuze/Guattari 1977**

Deleuze, Gilles; Guattari, Félix: Rhizom, Merve Verlag, Berlin 1977

**Deleuze/Guattari 1992**

Gilles, Deleuze; Felix, Guattari: Tausend Plateaus, Merve Verlag, Berlin 1992

**Dobbin 2012**

Dobbin, Claire: London Underground Maps. Art, Design and Cartography; Lund Humphries-Ashgate Publishing, Burlington-Farnham 2012

**Dotzler 2006**

Dotzler, Bernhard J.: Diskurs und Medium. Zur Archäologie der Computerkultur, Wilhelm Fink Verlag, München 2006

**Eastman u.a. 1974**

Eastman, Charles M. et al.: An Outline of the Building Description System. Research Report No.50 (RR-50), Carnegie-Mellon Univ., Pittsburgh 1974

**Eastman 1975**

Eastman, Charles M.; The Use of Computers instead of Drawings in Building Design, AIA Journal 63 (1975) 3, S.46-50

**Eastman u.a. 2008**

Eastman, Charles M. et al: BIM Handbook. A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, John Wiley & Sons, New Jersey 2008

**Eisenführ 2010**

Eisenführ, Franz; Weber, Martin; Langer, Thomas: Rationales Entscheiden, Springer, Heidelberg 2010

**Emmerer 2014**

Emmerer, Martin: Entwerfer am Mischpult – Neue Instrumente für den architektonischen Formfindungsprozess, in: GAM Magazine 10 (2014) S.55-67

**Engelbart 1962**

Engelbart, Douglas C.: Augmenting Human Intellect. A Conceptual Framework, Online unter: [http://www.doungengelbart.org/pubs/papers/scanned/Doug\\_Engelbart-AugmentingHumanIntellect.pdf](http://www.doungengelbart.org/pubs/papers/scanned/Doug_Engelbart-AugmentingHumanIntellect.pdf) 20.10.2014

**Ernst u.a. 2009**

Ernst, Gerhard; Steinbrenner, Jakob; Scholz, Oliver R.: From Logic to Art: themes from Nelson Goodman, Walter de Gruyter, Berlin 2009.

**Euler 1736 Solutio**

Euler, Leonhard: Solutio problematis ad geometriam situs pertinentis, The solution of a problem relating to the geometry of positions, in: Commentarii Academiae Scientiarum Imperialis Petropolitanae 8 (1736) , S.128-140.

**Euler 1736 Marinoni**

Euler, Leonhard: Brief an Giovanni Jacopo Marinoni vom 13. März 1736, PFA RAN: F.1, op.3, Nr.22, 18-18(ob)

**Fahrmeir u.a. 2011**

Fahrmeir, Ludwig; Künstler, Rita; Pigeot, Iris; Tutz, Gerhard: Statistik. Der Weg zur Datenanalyse, Springer, Berlin-Heidelberg 2011

**Flemming 1975**

Flemming, Ulrich: DIM — ein Programm zur Dimensionierung räumlicher Konfigurationen, Lehrveranstaltung Grundlagen systematischer Planung, Hochschulschrift, Berlin 1975

**Franck 2009**

Franck, Georg: Maschinelle Entwurfshilfen. Evolutionäre Algorithmen: der Ansatz der Künstlichen Kreativität, In: Gethmann, Daniel; Hauser, Susanne (Hg.): Kulturtechnik Entwerfen. Praktiken, Konzepte und Medien in Architektur und Design Science, transcript Verlag, Bielefeld 2009, S.234-241

**Frege 1879**

Frege, Gottlob: Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, Verlag Louis Nebert; Halle a/S 1879  
Online: <http://gallica.bnf.fr/ark:/12148/bpt6k65658c> 20.10.2014

**Friedman 1975**

Friedman, Yona: Towards a Scientific Architecture, MIT Press, Cambridge-London 1975

**Gabriel**

Gottfried, Gabriel: Merkmal in: Ritter, Joachim; Gründer, Karlfried (Hg.); Historisches Wörterbuch der Philosophie, Bd. 5, Schwabe&Co AG Verlag, Basel-Stuttgart 1980, Sp. 1153-1156

**Garcia 2010**

Garcia, Mark: The Diagrams of Architecture, John Wiley & Sons Ltd. Chichester 2010

**Gerdes u.a. 2004**

Gerdes, Ingrid; Klawonn, Frank; Kruse, Rudolf: Evolutionäre Algorithmen, Vieweg Verlag, Wiesbaden 2004

**Giesecke 1988**

Giesecke, Michael: Die Logik der Modellbildung bei A. Dürer In: Giesecke, Michael: Die Untersuchung institutioneller Kommunikation, Westdeutscher Verlag, Opladen 1988, S. 117-128



**Gigerenzer 2008**

Gigerenzer, Gerd: Bauchentscheidungen - Die Intelligenz, das unbewusste und die Macht der Intuition, Wilhelm Goldmann Verlag, München 2008

**Goodman 1985**

Goodman Nelson: How Buildings Mean, in: Critical Inquiry 11(1985), S.642-53

**Goodman 2012**

Goodman, Nelson: Sprachen der Kunst. Entwurf einer Symboltheorie, Suhrkamp Verlag, Frankfurt am Main 2012

**Gödel 1931**

Gödel, Kurt: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. In: Monatshefte für Mathematik und Physik 38(1931), S. 173-198

**Gottschalk 1968**

Gottschalk, Ottomar: Flexible Verwaltungsbauten. Planung Funktion Flächen Ausbau Einrichtung Kosten Beispiele, Verlag Schnelle Quickborn 1968

**Günzel 2007**

Günzel, Stephan: Topologie. Zur Raumbeschreibung in den Kultur- und Medienwissenschaften, transcript Verlag, Bielefeld 2007

**Günzel 2008**

Günzel, Stefan: Topologie und Städtischer Raum, in: der architekt 3 (2008), S.8-10

**Hanson 1998**

Hanson, Julienne: Decoding Homes and Houses, University Press, Cambridge 1998

**Hillier/Hanson 1984**

Hillier, Bill; Hanson, Julienne: The Social Logic of Space, University Press, Cambridge 1984

**Hillier 1996**

Hillier, Bill: Space is the Machine. A Configurational Theory of Architecture, University Press, Cambridge 1996

**Hirschberg 2014**

Hirschberg, Urs: "Augmented Architecture", Transparenz, Intuition und unsere Sinne, in: GAM Magazine 10 (2014), S.23-39

**Hogarth 2001**

Hogarth, Robin M.: Educating Intuition, University of Chicago Press, Chicago 2001

**Hovestadt 2013**

Hovestadt, Ludger: Cultivating the generic. A mathematically inspired pathway for Architecture, in: Hovestadt Ludger, Bühlmann Vera (Hg.): Eigenarchitektur - Computability as Literacy, Ambra Verlag, Wien 2013, S.9-67

**Huizinga 2011**

Huizinga, Johan: Homo Ludens. Vom Ursprung der Kultur im Spiel, Rowohlt Verlag, Hamburg 2011

**Kahn 1961**

Kahn, Louis I.: Architect. Alfred Newton Richards Medical Research Building, in: Museum of Modern Art Bulletin 28 (1961)1, S. 3-23.

**Kahneman 2011**

Kahneman, Daniel: Schnelles Denken, langsames Denken, Siedler Verlag, München 2011

**Kannheiser 1990**

Kannheiser, Werner: Methoden der Ingenieurpsychologie, in: Birbaumer, Niels; Frey, Dieter; Kuhl, Julius (Hg.): Enzyklopädie der Psychologie, Bd. 2, Hogrefe, Göttingen 1990, S.55-86

**Kant 1800**

Kant Immanuel: Logik — ein Handbuch zur Vorlesung, bei Friedrich Nicolovius, Königsberg 1800

**Kant 2011**

Kant, Immanuel: Kritik der reinen Vernunft (Erstauflage 1781), Anaconda Verlag, Köln 2011

**Kiemle 1967**

Kiemle, Manfred: Probleme der Architektur unter dem Aspekt der Informationsästhetik, Verlag Schnelle, Quickborn 1967

**Kruse et al 1993**

Kruse, Rudolf; Gebhardt, Jörg; Klawonn, Frank: Fuzzy-Systeme, Teubner, Stuttgart 1993

**Leibniz 1996**

Leibniz, Gottfried Wilhelm: Hauptschriften zur Grundlegung der Philosophie, Bd.1, Felix Meiner Verlag, Hamburg 1996

**Levine 1982**

Levine, Neil: The Competition for the Grand Prix in 1824. A Case Study in Architectural Education at the Ecole Des Beaux-arts, in: Middleton Robin (Hg.): The Beaux-Arts and Nineteenth-Century French Architecture, MIT Press, Cambridge 1982, S.67-123

**Liu u.a. 1992**

Liu, Y.; Arimoto, S.: Path Planning using a Tangent Graph for Mobile Robots among Polygonal and Curved Obstacles. In: The International Journal of Robotics Research, 11 (1992), S. 376-382

**Marks 1974**

Marks, Lawrence E.: Sensory Processes: The New Psychophysics. Academic Press, New York 1974

**Meyer 2005**

Meyer, Eric: Cascading Style Sheets. Das umfassende Handbuch, O'Reilly, Köln 2005

**Mitchell 1977**

Mitchell, William John: Computer-Aided Architectural Design, Petrocelli Charter, New York 1977

**Mitchell 1996**

Mitchell, William John: The Logic of Architecture — Design, Computation, and Cognition, The MIT Press, Cambridge-London 1996

**Moriyama 1971**

Moriyama Raymond, Architects and Planners: Metropolitan Toronto Central Library programme and site selection study, Toronto 1971

**Murphy 2005**

Murphy, Michael D.: Landscape Architecture Theory. An evolving Body of Thought, Waveland Press, Long Grove 2005

**Murphy 2006**

Murphy, Michelle: Sick Building Syndrome and the Politics of Uncertainty, Duke University Press, Durham 2006

**Mühlbacher u.a. 2013**

Mühlbacher, Axel C; Kaczynski, Anika: Der Analytic Hierarchy Process (AHP). Eine Methode zur Entscheidungsunterstützung im Gesundheitswesen, in: PharmacoEconomics. German Research Articles, 11 (2013) 2, S. 119-132

**Müller-Herbers 2007**

Müller-Herbers, Sabine: Methoden zur Beurteilung von Varianten, [www.igp.uni-stuttgart.de/lehre/1\\_Vorlesung/download/Reader/Methoden.pdf](http://www.igp.uni-stuttgart.de/lehre/1_Vorlesung/download/Reader/Methoden.pdf) in: [www.igp.uni-stuttgart.de](http://www.igp.uni-stuttgart.de) 20.10.2014

**Negroponte/Weinzapfel 1976**

Negroponte, Nicholas; Weinzapfel, Guy: Architecture-by-Yourself. An Experiment with Computer Graphics for House Design, in: SIGGRAPH (Hg.): Proceedings of the 3rd annual conference on Computer graphics and interactive techniques, ACM, New York 1976 S.74-78

**Narens 1968**

Narens, L.; Luce, R. D.: Measurement. The Theory of Numerical Assignments, in: Psychological Bulletin 99 (1986) H.2, S. 166—180

**Orth 1974**

Orth, B.: Einführung in die Theorie des Messens. Kohlhammer, Stuttgart-Berlin-Köln 1974

**Pawlowski 1980**

Pawlowski, Tadeusz: Begriffsbildung und Definition, de Gruyter, Berlin 1980.

**Peña/Parshall 2001**

Peña, William; Parshall, Steven: Problem Seeking: An Architectural Programming Primer, Wiley, London 2001

**Pinzger u.a. 2005**

Pinzger, Martin; Gall, Harald; Fischer, Michael; Lanza, Michele: Visualizing Multiple Evolution Metrics, in: ACM (Hg.): Proceedings of SoftVis 2005 (Proceedings of the 2005 ACM Symposium on Software Visualization), ACM Press, New York 2005 S.67-75

**Poschard 1995**

Poschardt, Ulf: DJ Culture, Hamburg 1995

**Ratti 2003**

Ratti, Carlo: Urban texture and space syntax: some inconsistencies, in: Environment and Planning 31 (2003) H.4, S.501-511

**Rasch u.a. 2010**

Rasch, Björn; Friese, Malte; Hofmann, Wilhelm; Naumann, Ewald: Quantitative Methoden. Einführung in die Statistik für Psychologen und Sozialwissenschaftler, Springer, Berlin-Heidelberg 2010

**Reber 2009**

Reber, Franz: Vitruv – Zehn Bücher über Architektur. De Architectura Libri decem. matrixverlag, Wiesbaden 2009

**Rittel/Webber 1973**

Rittel, Horst W.J.; Webber, Melvin M.: Dilemmas in a General Theory of Planning, in: Policy Sciences 4 (1973), S.155-169

**Samuel 2010**

Samuel, Flora: Le Corbusier and the architectural promenade, Birkhäuser Verlag, Basel 2010

**Saussure 2001**

Saussure, Ferdinand de: Grundfragen der allgemeinen Sprachwissenschaft, de Gruyter, Berlin 2001

**Schmeing/Kleinheinz 1999**

Schmeing, Astrid; Kleinheinz, Lena: Zum Begriff des Entwerfens, in: Wolkenkuckucksheim. Internationale Zeitschrift zur Theorie der Architektur, 4 (1999) H. 1, Online unter: <http://www.cloud-cuckoo.net/openarchive/wolke/deu/Themen/991/Schmeing/schmeing.html> 20.10.2014

**Stachowiak 1973**

Stachowiak, Herbert: Allgemeine Modelltheorie, Springer, Wien-New York 1973.

**Staud 2005**

Staud, Josef L.: Datenmodellierung und Datenbankentwurf - Ein vergleich aktueller Methoden, Springer, Berlin-Heidelberg 2005

**Steadman 1983**

Steadman, J.P.: Architectural Morphology. An Introduction to the Geometry of Building Plans, Pion Limited, London 1983

**Stevens 1946**

Stevens, Stanley Smith: On the Theory of Scales of Measurement, in: Science 103 (1946) S.677-680

**Strassert 1995**

Strassert, Günter: Das Abwägungsproblem bei multikriteriellen Entscheidungsproblemen. Grundlagen und Lösungsansatz unter besonderer Berücksichtigung der Regionalplanung, Peter Lang, Frankfurt am Main 1995

**Taylor 2012**

Taylor, Mark: Diagramming the Interior, in: Garcia, Mark (Hg.): The Diagrams of Architecture, Wiley, London 2012, S.134-141

**Theuer 1912**

Theuer, Max: Die zehn Bücher über die Baukunst, Heller, Wien-Leipzig 1912

**Tidwell 2002**

Tidwell, Doug: XSLT, O'Reilly, Peking-Cambridge-Köln 2002

**Timerding 1919**

Timerding, Heinrich Emil: Kant und Euler, in: Kant-Studien 23 (1919) H. 1-3, S.18-64

**Thompson 1989**

Thompson, Godfrey: Library Buildings, Butterworth Architecture, London-Boston-Singapore 1989

**Tufte 1990**

Tufte, Edward R.: Envisioning Information. Narratives of Space and Time, Graphics Press LLC, Cheshire 2008 [1. Auflage 1990]

**Turing 1950**

Turing, Alan Mathison: Rechenmaschinen und Intelligenz, in: Dotzler, B.J., Kittler F.: Intelligence Service, Schriften von Alan Turing, Brinkmann & Bösse, Berlin 1987, S.147-114

**Tügel 1968**

Tügel, Peter W.: Quickborner Team, in: ARCH+ 2 (1968), S.9-13

**Valena u.a. 2011**

Valena, Tomás; Avermaete, Tom; Vrachliotis, Georg: Structuralism Reloaded. Rule-Based Design in Architecture and Urbanism, Menges Verlag, Stuttgart-London 2011

**Velminski 2009**

Velminski, Wladimir: Leonhard Euler - Die Geburt der Graphentheorie, Kulturverlag Kadmos, Berlin 2009

**Vrachliotis 2012**

Vrachliotis, Georg: Geregelte Verhältnisse, Springer, Wien-New York 2012

**Vonhoegen 2011**

Vonhoegen, Helmut: Einstieg in XML - Grundlagen, Praxis, Referenz, Galileo Press, Bonn 2011

**Wittgenstein 2003**

Wittgenstein, Ludwig: Philosophische Untersuchungen, Suhrkamp Verlag, Frankfurt am Main 2003

**Zadeh 1965**

Zadeh, Lotfi A.: Fuzzy sets, in: Information and Control 8 (1965), S.338-353

**Zangemeister 1970**

Zangemeister, Christoph: Nutzwertanalyse in der Systemtechnik. Eine Methode zur multidimensionalen Bewertung und Auswahl von Projektalternativen, Wittmannsche Buchhandlung, München 1970

## Online Quellen

### Duden online

[www.duden.de](http://www.duden.de)

### Encyclopedia of Mathematics

[www.encyclopediaofmath.org](http://www.encyclopediaofmath.org)

### Stanford Encyclopedia of Philosophy

[plato.stanford.edu](http://plato.stanford.edu)

### Wirtschaftslexikon

[www.onpulson.de](http://www.onpulson.de)

### Wikiwörterbuch Wiktionary

[www.wiktionary.org](http://www.wiktionary.org)

## Normen

### ISO 16739

Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries

### ISO10303-28

XML representation of EXPRESS schemas and data

### ISO 9699

Performance standards in building — Checklist for briefing — Contents of brief for building design

### DIN 18205

Bedarfsplanung im Bauwesen

# ABBILDUNGEN



## Einleitung

0.1	Saussure um 1910, Das Sprachliche Zeichen, in: Saussure 2001, 78.	5
0.2	TARSKI'S WORLD 1987: Screenshot 2008 © CSLI Publications Leland Stanford Junior University	9
0.3	TARSKI'S WORLD: Verfügbares Vokabular an Satzfunktionen	10
0.4	Delphi Code: Messfunktionen der Prädikate <i>Cube</i> und <i>LeftOf</i>	10

## 1 Raumprogramme – Architektur à la Carte?

1.1	Raumprogramm für den Entwurf eines Kindergartens: Beilage zum Architekturwettbewerb Kindergarten im Stadtpark (Auszug), Stadt Wien 2009, Magistratsabteilung 19, Architektur und Stadtgestaltung	20
1.2	Christine Frederick 1914, Prozesskette zwischen Küche und Speiseraum, in: Garcia 2010, 138.	23
1.3	Thompson 1989: Beziehungsmatrix der Funktionsbereiche einer Bibliothek in: Thompson 1989, 41.	25
1.4	Le Corbusier 1945, Strukturdiagramm Unité d'habitation, in: Garcia 2010, 15.	26
1.5	(a) Wright um 1940, Drei Entwurfslösungen mit gleicher Topologie (b) Mitchell 1990, Strukturdiagramm, in: Mitchell 1996, 119f.	27
1.6	Friedman 1975, Unlösbare Entwurfsaufgabe, in: Friedman 1975, 39.	29

## 2 Graphen – Lernen von Königsberg

2.1	Matthäus Merian um 1640: Kupferstich (38X29cm), Königsberg in der Vogelschau, in: Biggs 1998, 2.	32
2.2	Euler 1736: Diagramm der Stadt Königsberg, in: Euler 1736, 129.	33
2.3	Rechnung	35
2.4	(a) Euler 1736: Skizze mit dem Vorschlag einer achten Brücke „h“. (b) Rechnung, in: Günzel 2007, 180.	35
2.5	Stadt Königsberg als Graph © M. Emmerer, 2014	36
2.6	Graph und Dualer Graph © M. Emmerer, 2014	37
2.7	Isomorphe Graphen © M. Emmerer, 2014	38
2.8	Graph in Matrixform © M. Emmerer, 2014	38
2.9	Unbekannter Künstler 1908: Londoner U-Bahn Netz, in: Dobbin 2011, 63.	40
2.10	Harry Beck 1933, Londoner U-Bahn Netz, in: Dobbin 2011, 81.	40
2.11	Czechoslovakia Air Transport Company 1933, Flugplan, in: Tufte 1990, 102.	41
2.12	Adjacency- und Access Graph desselben Grundrisses © M. Emmerer, 2014	43
2.13	Steadman 1983, Grundriss und (justified) Access Graph eines Gebäudes, in: Steadman 1983, 230.	43

2.14	Quickborner Team um 1970, Informationsfluss in einem Großraumbüro, in: Murphy 2006, 35ff.	44
2.15	Gottschalk 1968, Beziehungen zwischen Arbeitsplätzen, in: Gottschalk 1968, 183 u. 247.	45
2.16	Kantowitz & Sorkin 1983, Verknüpfungsanalyse Küche: in: Kannheiser 1990, 74.	46
2.17	Tomor Elezkurtay 2002, Screenshot Floor Plan Program: in: Valena u.a. 2011, 278.	46

### 3 Formalisierte Gebäudespezifikation

3.1	Merkmalstypen © M. Emmerer, 2014	52
3.2	Nominalskala © M. Emmerer, 2014	54
3.3	Ordinalskala © M. Emmerer, 2014	54
3.4	Intervallskala © M. Emmerer, 2014	55
3.5	Verhältnisskala © M. Emmerer, 2014	55
3.6	Absolutskala © M. Emmerer, 2014	55
3.7	Transformation von der Verhältnisskala ( $\mathfrak{R}^+$ ) zur Ordinalskala (S,M,L) © M. Emmerer, 2014	56
3.8	Beispiel Wertefunktion Raumgröße © M. Emmerer, 2014	58
3.9	Beispiel Wertefunktion Belichtung © M. Emmerer, 2014	58
3.10	Beispiel Wertefunktion Figur © M. Emmerer, 2014	58
3.11	„Weiche“ Ungleichung © M. Emmerer, 2014	59
3.12	Mehrere Hochpunkte © M. Emmerer, 2014	59
3.13	Gruppen	61
3.14	Räume und Raumzonen	61
3.15	Relationen	62
3.16	Qualitative Merkmale	63
3.17	Quantitative Merkmale	63
3.18	Qualitative Präferenzfunktion	64
3.19	Qualitative (diskrete) Präferenzfunktion © M. Emmerer, 2014	64
3.20	Quantitative Präferenzfunktion	64
3.21	Quantitative (stetige) Präferenzfunktion © M. Emmerer, 2014	64
3.22	unstetige Präferenzfunktion © M. Emmerer, 2014	65
3.23	Gebäudespezifikation in XSAL-Notation	65
3.24	XML NOTEPAD © Microsoft	67
3.25	Kiviat Graph, in: Pinzger u.a. 2005, 73.	68
3.26	Martin Emmerer 2014, Design Goal Manager © M. Emmerer, 2014	69
3.27	Slider und XSAL-Tag für Merkmale unterschiedlicher Skalenniveaus © M. Emmerer, 2014	70
3.28	Screenshot CURVE 2, Software Synthesizer mit Waveform Editor © www.cableguys.de	71



## 4 Building Information Model

4.1	Martin Emmerer 2009, DRAFTPAD © M. Emmerer, 2009	78
4.2	Martin Emmerer 2009, Screenshots DRAFTPAD © M. Emmerer, 2009	79
4.3	Entitäten und Attribute © M. Emmerer, 2014	80
4.4	Delphi Code: Datenobjekte	81
4.5	Martin Emmerer 2009, Screenshot DRAFTPAD, Adjacency- und Access Graph © M. Emmerer, 2009	82
4.6	Access Graph vs. Fluchtweg aus Raum 3	83
4.7	Verschiedene Methoden zur Bildung kürzester Wege innerhalb eines Polygons. (a) Rasterung (b) Triangulation (c) Visibility-Graph	84
4.8	Martin Emmerer 2009, Screenshot DRAFTPAD, Visibility Graph © M. Emmerer, 2009	85
4.9	(a) Konfigurationsraum, (b) Abstraktion des menschlichen Körpers © M. Emmerer, 2014	86
4.10	Delphi Code: Datenobjekte Graph, Node, Edge	86
4.11	Delphi Code: Objektstruktur Gebäudedatenmodell	87

## 5 Algorithmische Explikation

5.1	Makrostruktur der Algorithmische Explikation © M. Emmerer, 2014	89
5.2	Funktionen für 4 Skalenniveaus	94
5.3	(a) Drei Entwurfslösungen (b) Merkmalsvektoren © M. Emmerer, 2014	95
5.4	Programmfunktion <i>Window Quotient</i>	95
5.5	Programmfunktion <i>Order of Magnitude</i>	96
5.6	Formular-Layout Architektur Routine © M. Emmerer, 2014	97
5.7	Übersicht Algorithmische Explikation © M. Emmerer, 2014	98
5.8	Merkmalskatalog	99

## 6 Anwendungen

6.1	abstrakte Klasse © M. Emmerer, 2014	100
6.2	Präferenzfunktion, Ermittlung des Zielerfüllungsgrades © M. Emmerer, 2014	103
6.3	Pseudocode Funktion <i>Satisfaction</i>	103
6.4	XSAL Spezifikation	103
6.5	(a) Entwurfslösung (b) Merkmalsvektor © M. Emmerer, 2014	104
6.6	(a) Konstruktion <i>Satisfaction</i> (b) Satisfaktionsvektor (c) Eigenschaftsprofil © M. Emmerer, 2014	105
6.7	Prioritäten-Mixer © M. Emmerer, 2014	107
6.8	(a) Entwurfslösung (b) Spezifikation in Tabellenform © M. Emmerer, 2014	108
6.9	(a) Fitnesswerte (b) Berechnung Gesamtfitness $w_{1-n}=1/8$ © M. Emmerer, 2014	109
6.10	Fitnessprofil © M. Emmerer, 2014	110
6.11	Vier Entwurfslösungen © M. Emmerer, 2014	111
6.12	Vier Fitnessprofile © M. Emmerer, 2014	111
6.13	ARCHILL.ES Programmkomponenten © M. Emmerer, 2014	113

6.14	Martin Emmerer 2014, DRAFTPAD und DESIGN GOAL MANAGER © M. Emmerer, 2014	114
6.15	ARCHILL.ES: Von der Topologie zur Geometrie © M. Emmerer, 2014	115
6.16	ARCHILL.ES: Von der Geometrie zur Topologie © M. Emmerer, 2014	116
6.17	Screenshots ARCHILL.ES (a) Orthogonalität, (b) Nettonutzflächen, (c) Integration © M. Emmerer, 2014	117
6.18	Screenshot ARCHILL.ES Qualität der Orientierung von Wohneinheiten © M. Emmerer, 2014	118
6.19	Screenshots ARCHILL.ES, (a) Gebäudespezifikation als Graph und unter- schiedlicher Entwurfslösungen, (b) Entwurf und Fitnessprofil © M. Emmerer, 2014	119
6.20	Screenshot ARCHILL.ES: Facettierte Suche im Building Information Model © M. Emmerer, 2014	120

## Conclusion – Der Kreis schließt sich

7.1	Translation: Bildung von Vektoren © M. Emmerer, 2014	123
-----	--	-----

# CURRICULUM VITAE



## Personal

Name            Martin Emmerer  
1976            born in Graz  
                  Austrian Nationality

## Professional

2000            Foundation ERSCHEINUNGSBILD Visualization  
2006            Foundation HoG architektur ZT GmbH  
2006 -           Lecturer, University of Technology Graz

## Education

1995            Matura (Modern Languages and Latin) Gymnasium Sacré Coeur Graz  
1996 - 2006    Architecture Studies  
2006            Architecture Diploma University of Technology Graz  
2010 - 2014    Doctoral School of Architecture, University of Technology Graz

## Buildings

2009            New Southern Wing - Castle Museum Linz  
2009 - 2014    Modification of the Franciscan monastery in Graz  
2012            Residential building *am Steinberg*  
2013            Townhouse *Ballhaus* Graz

## Languages

German, English, French