GAM.

ARCHITECTURE MAGAZINE 10

Intuition & the Machine

AMBRA | V

TU Graz

BUILDING DESIGN CULTURE ART THEORY HISTORY LANDSCAPE PRACTICE ECOLOGY PLANNING TECHNOLOGY URBANISM

2 **Intuition & the Machine**

3 **Editorial**

6 **Senses & Tools**

8 Makoto Sei Watanabe
ALGOrithmicDesign/Can the Child Machine Develop into an Adult?
ALGOrithmischesDesign. Kann die Kind-Maschine erwachsen werden?

22 Urs Hirschberg
„Augmented Architecture“ Transparenz, Intuition und unsere Sinne
“Augmented Architecture:” Transparency, Intuition and Our Senses

40 Sean Hanna
Machine Intuition

54 Martin Emmerer
Entwerfer am Mischpult. Neue Instrumente für den architektonischen Formfindungsprozess
Designer at the Mixing Console: New Instruments for the Architectural Form-Generation Process

68 Kristof Crolla
Golden Moon: Digital Control Meets Craftsmanship **Golden Moon. Digitale Kontrolle trifft Handwerkskunst**

76 Interview: Edith Ackermann/Urs Hirschberg
Embodied Cognition and Reflective Practices **Verkörperte Kognition und reflexive Verfahren**

88 **Digital Grotesque | Michael Hansmeyer & Benjamin Dillenburger**

98 **Mind & Matter**

100 Mario Carpo
Digital Phenomenologies: From Postmodern Indeterminacy to Big Data and Computation
Digitale Phänomenologien. Von postmoderner Unbestimmtheit zu Big Data und Kalkulation

112 Nathalie Bredella
Die Wissenspraktiken der *Paperless Studios* **The Knowledge Practices of the “Paperless Studio”**

128 George L. Legendre
God Spare Me Intuition (and I’ll Take Care of the Machine Myself)
Gott verschone mich mit Intuition (und um die Maschine kümmere ich mich selbst)

142 Chris Younès
Between Machine and Intuition, Thresholds and Hybridizations
Zwischen Maschine und Intuition, Schwellen und Hybridbildungen

152 Interview: Ludger Hovestadt/Urs Hirschberg
„Einfachheit ist für Anfänger“ **“Simplicity is for Beginners”**

170 **Decibot, Fluid Crystallization | Skylar Tibbits**

182 **Interactions & Mutations**

184 Felix Raspall
Digital-Material Feedback: Extending Intuition into the Production Process
Digital-materielles Feedback: Zur Ausdehnung der Intuition auf den Produktionsprozess

196 Kathrin Dörfler • Romana Rust • Florian Rist
Moderation der Unschärfe. Experimente zur Verschränkung physischer und digitaler Formfindungsprozesse
Moderation of Vagueness: Experiments on the Interconnection between Physical and Digital Processes of Form Generation

206 Tobias Wallisser
Parametrische Adaption. Eine Evolution des Entwurfsprozesses
Parametric Adaption: An Evolution of the Design Process

222 François Roche
An Architecture *des Humeurs*

236 Ammar Mirjan • Jan Willmann • Fabio Gramazio • Matthias Kohler
Designing Behavior: Materializing Architecture with Flying Machines
Die Gestaltung von maschinellem Verhalten. Architekturmaterialisierung mit Flugmaschinen

248 Interview: Skylar Tibbits/Urs Hirschberg
“Surprise is almost everything in design!” **„Überraschung ist beim Entwerfen fast alles!“**

266 **AutorInnen** **Authors**

272 **Bücher** **Books**

304 **Aus der Fakultät** **Faculty News**

342 **Call for Papers GAM.11**

1 DJ at Work, 2013 © photodune.net

Entwerfer am Mischpult

Neue Instrumente für den architektonischen Formfindungsprozess

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 (Abb. 1)!

Designer at the Mixing Console: New Instruments for the Architectural Form-Generation Process. Deft hands are moving the slider and turning knobs. Parameters are defined and values set. At the same time, a spatial structure is generated that changes with each adjustment of a variable. Forms develop as controlled, metastable phenomena. Propelled by interaction, a three-dimensional visual aspires to self-organization. The setup stands. The room is quaking. With the Unité d'Habitation as baseline, accompanied by a nicely tuned mix of variables, the house DJ is at work (fig. 1)!

MARTIN EMMERER

Schon die Arbeiten von Marcel Duchamp und Andy Warhol haben gezeigt, dass der Einsatz neuer Technologien in der Kunst und die damit häufig einhergehende Reproduzierbarkeit von Kunstwerken nicht die Auslöschung des Status des Künstlers als autonomes Schöpfersubjekt bedeuten muss. Der Prozess der Modernisierung des Konzepts des Künstlers als Autor hatte bereits in der ersten Hälfte des 20. Jahrhunderts begonnen. Auch in der Musik kam es durch die zunehmende Technisierung zu grundlegenden Veränderungen. Zahlreiche Musiker legten ihre alten Instrumente beiseite und begannen stattdessen mit Beatboxen, Samplern und Mischpulten zu experimentieren. Wie die Pop Art der 1960er Jahre nahm auch die DJ-Kultur in der zweiten Hälfte des 20. Jahrhunderts einen ambivalenten Platz zwischen der Dekonstruktion und der Bewahrung der Autonomie des Künstlers ein.[1] Durch die Technik des Samplings verschwindet der DJ als Autor hinter seinem elektronischen Equipment und scheint durch seine referenzielle Praxis, also dem maßlosen Umgang mit den Werken anderer Musiker, der Funktion der künstlerischen Autonomie jeden Respekt zu verweigern.[2] Dennoch wird der DJ heute als eigenständiger Künstler wahrgenommen – wenn auch auf einer erweiterten Ebene. Matt Black, Mitbegründer des britischen Musikproduzentenduos Coldcut bemerkte dazu selbstbewusst: „We see ourselves as cut creators, scratching, sampling and mixing is basically the same thing for us as guitarplaying for Johnny Winter or Jeff Healy.“[3] Und genau wie bei Gitarrenspielern oder Malern gibt es in jedem Bereich Amateure und Experten: „It’s exactly the same with the sampler. Either you use it as a toy or you become a virtuoso.“[4]

Zur Festlegung der erforderlichen Größe jedes einzelnen Raumes entwarf Negroponte eine Art digitales Mischpult.

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 bzw. „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.

1 Vgl. Ulf Poschardt: *DJ Culture*, Hamburg 1995, S. 2.
2 Vgl. ebd., S. 375f.
3 Ebd.
4 Ebd.

The work of Marcel Duchamp and Andy Warhol already showed that the implementation of new technology in art, along with the often concomitant reproducibility of artwork, need not signify the obliteration of the artist status as autonomous creator-subject. The process of modernizing the concept of the artist as author already began in the first half of the twentieth century. In music, fundamental changes were seen as well, thanks to advancing mechanization. Countless musicians put away their old instruments to instead start experimenting with beat boxes, samples, and mixing consoles. Like the Pop Art of the 1960s, DJ culture assumed an ambivalent position between deconstruction and a desire to retain the autonomy of the artist in the second half of the twentieth century.[1] With the advent of sampling technology, the DJ as author disappeared behind electronic equipment, and through his referential practice, that is, the extortionate approach to treating the works of other musicians, he has turned his back to the function of artistic autonomy in all respects.[2] Nonetheless, today DJs are perceived to be artists in their own right, though in a broader sense. As Matt Black, cofounder of the British music-producer duo Coldcut, self-confidently observed on this topic: “We see ourselves as cut creators, scratching, sampling and mixing is basically the same thing for us as guitarplaying for Johnny Winter or Jeff Healy.”[3] And just as with guitar players or painters, there are amateurs and experts in any field: “It’s exactly the same with the sampler. Either you use it as a toy or you become a virtuoso.”[4]

Since the 1960s, the architectural computer avant-garde has been dreaming of transferring form-generation processes to the machine. The idea of viewing architecture as a self-creating, autopoietic system in which humans intervene only from the outside by determining or “mixing” parameters has fascinated since the very beginnings of structuralism. Cybernetics, autonomous architecture, and self-organization are catchphrases that are frequently used in this context. The motives for such visions are manifold. In any case, they mirror the hope that computers might surprise us with breathtaking new spatial complexes—generating buildings that we would never have been able to achieve on our own. Some architects would even be willing to surrender part of their direct control over form in the name of realizing this idea.

An important contribution to the human-machine debate was made in the year 1967 by the Architecture Machine Group, founded by Nicholas Negroponte at the Massachusetts Institute of Technology. The possibilities offered by the computer as an

1 See Ulf Poschardt, *DJ Culture* (Hamburg, 1995), p. 2.
2 See ibid., pp. 375–76.
3 Ibid.
4 Ibid.

instrument of design composition were to be explored through playful experiments and art installations. The ambitious goal was to develop machines that go beyond simplifying architects' design activity to instead completely take over their tasks. Negroponte was convinced that it would be easier to spark the enthusiasm of architects for innovative technology via the path of practice over that of theory.[5] For example, he felt that designers shouldn't have to learn how to deal with unfamiliar computer codes. Special focus was placed on the development of suitable interfaces. This extended beyond the purely visual interaction between human and machine, as had been seen with Ivan Sutherland's Sketchpad. The planning program URBAN5 of 1967 was already designed in such a way that users could communicate with the computer not only visually but also by voice.[6] "It should be possible to enter the concrete questions, considerations, and directives related to a planning project, such as the optimal position of a house entrance or windows, in the form of a complete data set. The computer would understand the meaning of this data and calculate solutions accordingly, field counter-questions, and even make recommendations as to what should be done next or remembered during the planning process."[7]

In the year 1976, Negroponte developed a computer program called YONA, which reflected the implementation of a design methodology created by Yona Friedman.[8] The design process took place in four consecutive steps. First, the user had to define which rooms were to be allocated to his building, and he could choose from a predefined list of rooms. If a desired room was not available on the list, it could be manually entered by typing in the appropriate specifications. Negroponte developed a kind of digital mixing console for determining the necessary size of each individual room. Aided by slide controls, which spanned an interval of "very small" and "very large," the respective required area could be regulated. When the "area arrow" was moved into a new position, a numerical value was updated next to the slide control. Operating in the background was a "background routine" that calculated the total required floor space and offered a prognosis of building costs. The second step involved inputting the desired accessibility options between the rooms. Each added connection between two rooms caused a proximity matrix to be updated, which clearly showed the user the "links" already defined between specific rooms. The program used the data entered in this way to automatically

Einen wichtigen Beitrag in der Mensch-Maschine Debatte leistete die im Jahre 1967 von Nicholas Negroponte am Massachusetts Institute of Technology gegründete Architecture Machine Group. In spielerischen Experimenten und Kunstinstallationen sollte der Möglichkeitsraum des Computers als Instrument der gestalterischen Komposition erkundet werden. Das ehrgeizige Ziel bestand in der Entwicklung von Maschinen, welche die entwerferische Tätigkeit des Architekten nicht nur erleichtern, sondern sie ihm sogar vollständig abnehmen sollten. Für Negroponte stand fest, dass man Architekten für neuartige Technologien besser über den Weg der Praxis als über jenen der Theorie begeistern konnte.[5] Entwerfer sollten nicht etwa den ungewohnten Umgang mit Computercodes lernen müssen. Ein besonderes Augenmerk galt der Entwicklung geeigneterer Schnittstellen. Dabei ging man über eine rein visuelle Interaktion zwischen Mensch und Maschine im Sinne von Ivan Sutherlands Sketchpad hinaus. Bereits bei dem 1967 entwickelten Planungsprogramm URBAN5 sollte der User nicht nur visuell, sondern auch sprachlich mit dem Computer kommunizieren.[6] „Konkrete Fragen, Überlegungen und Anweisungen zu einem Planungsprojekt, etwa nach der optimalen Lage eines Hauseingangs oder der Fenster, sollten in Form von ganzen Sätzen eingegeben werden können. Der Computer würde die Bedeutung dieser Sätze verstehen und dementsprechend Lösungen berechnen, Gegenfragen formulieren und sogar Empfehlungen aussprechen, was im Planungsprozess als nächstes zu tun sei oder nicht vergessen werden dürfte." [7]

Im Jahre 1976 entwickelte Negroponte das Computerprogramm YONA, welches die Implementierung einer von Yona Friedman entwickelten Entwurfsmethodik darstellte.[8] Der Entwurfsprozess erfolgte in vier aufeinanderfolgenden Schritten: Als erstes musste der User definieren, welche Räume in sein Gebäude eingeplant werden sollten. Dazu konnte aus einer vordefinierten Liste von Räumen gewählt werden. War einer der gewünschten Räume nicht darin enthalten, so erfolgte die Eingabe durch das Eintippen der Raumbezeichnung. Zur Festlegung der erforderlichen Größe jedes einzelnen Raumes entwarf Negroponte eine Art digitales Mischpult. Mithilfe von Schiebereglern, welche ein Intervall zwischen „very small" und „very large" aufspannten, konnte der jeweilige Flächenbedarf reguliert werden. Während man den „area arrow" auf eine neue Position verschob, aktualisierte sich ein numerischer Wert neben dem Schieberegler. Im Hintergrund arbeitete eine „Background Routine", welche den Gesamtflächenbedarf berechnete und eine Prognose über die Baukosten lieferte. Im zweiten Schritt erfolgte die Eingabe der gewünschten Zugänglichkeiten der Räume untereinander. Das Hinzufügen jeder Verbindung zwischen zwei Räumen aktualisierte eine Nachbarschaftsmatrix, welche dem User übersichtlich vor Augen führte, zwischen welchen Räumen bereits ein „Link" definiert wurde. Aus den auf diese Weise eingegebenen Daten generierte das Programm automatisch einen zweidimensionalen (planaren) Graphen, dessen Knoten vom User manuell an eine neue Position geschoben werden konnten.

5 See Georg Vrachliotis, *Geregelte Verhältnisse* (Vienna, 2012), pp. 226ff.

6 See Nicholas Negroponte, "Urban5," in *The Architecture Machine* (Cambridge, MA, 1970), pp. 70–93.

7 Vrachliotis, *Geregelte Verhältnisse*, pp. 209–10 (see note 5).

8 See Yona Friedman, *Toward a Scientific Architecture* (Cambridge, MA, 1975).

5 Vgl. Georg Vrachliotis: *Geregelte Verhältnisse*, Wien 2012, S. 226ff.

6 Vgl. Nicholas Negroponte: „Urban5", in: *The Architecture Machine*, Cambridge, MA 1970, S. 70–93.

7 Vrachliotis: *Geregelte Verhältnisse*, S. 209f. (wie Anm. 5).

8 Vgl. Yona Friedman: *Towards a Scientific Architecture*, Cambridge, MA 1975.

Während der Graph auf diese Weise vom User manipuliert wurde, arbeiteten im Hintergrund erneut Computerroutinen, welche das topologische Arrangement evaluierten und dem Benutzer Fehlermeldungen ausgaben, wenn diese für die spätere geometrische Anordnung der Räume Konflikte prognostizierten. Zum damaligen Zeitpunkt beschränkte sich das System auf nur zwei Warnungen: „No single level arrangement can be found for this set of links“ und „[n]o arrangement is possible without some enclosed space“. Dies bedeutete, dass die räumliche Umsetzung des Arrangements auf Grund der gewünschten Raumbeziehungen entweder auf mehr als einer Ebene erfolgen musste, oder das Einfügen von zusätzlichen Zwischenräumen, wie etwa einem Korridor, notwendig machte. Im Falle einer Erweiterung des Systems durch zusätzliche „Background Routines“ könnte in Zukunft die Ausrichtung der Räume, so die Autoren, auch auf Aussicht oder Lichteinfall etc. überprüft werden. Nachdem durch die Manipulation des Graphen durch den User eine befriedigende Dateneingabe erreicht wurde, d.h., vom System keine Fehlermeldungen mehr angezeigt wurden, konnte der dritte Schritt eingeleitet werden: Die Umrechnung der topologischen Struktur in ein Nurbs-Modell. Die Fläche jeder einzelnen Blase entsprach bereits exakt den vorher definierten Wünschen. Im letzten Schritt sollte dem User das zellenartige Gebilde als Unterlage für den manuellen Entwurf der endgültigen Gestalt der Räume dienen. Die auf dem Monitor handgezeichneten Konturen wurden im Hintergrund ebenfalls maschinell überwacht und maßgebliche Abweichungen von den vorher definierten Flächenanforderungen angezeigt. War auch dieser Vorgang vollständig abgeschlossen, konnten die Blasen gelöscht werden.

Im Fazit des Aufsatzes „Architecture-by-Yourself“ (1976) von Negroponte und Weinzapfel wird das eben vorgestellte Konzept von den Autoren selbst abwertend als eine „worst-case exercise in man-machine interaction“[9] bezeichnet. Die Architecture Machine Group sah ihr Ziel der Verbindung von Mensch und Maschine im architektonischen Entwurfsprozess durch dieses Experiment offensichtlich nicht erfüllt. Auch für Yona Friedmann war das Ergebnis des Experiments ernüchternd: „My experiment with MIT made me understand a fundamental disadvantage of using computers for design: they are too fast. People have their own thinking speed and computer thinking is too fast, too intransparent for them.“[10]

Die Architektur ist nicht die einzige Disziplin, in der man mit komplexen Problemen konfrontiert ist und die nach neuen Wegen der Computerunterstützung sucht. In verschiedenen Bereichen der Wissenschaft, wie etwa der Molekularbiologie, werden bereits Softwaretools entwickelt, mit deren Hilfe es gelingt, die menschliche Intuition zu nutzen, um komplexe kombinatorische Probleme zu lösen. Wo selbst hochentwickelte Computeralgorithmen versagen, brillieren Menschen durch ihre natürlich gegebenen Fähigkeiten. Eines dieser Tools ist FOLD.IT (Abb. 2). Diese Software – kreiert von dem Molekularbiologen David Baker und dem Informatiker und Spiele-Entwickler Zoran Popović – nutzt das herausragende räumliche Vorstellungsvermögen des Menschen für wissenschaftliche Zwecke: Der

9 Nicholas Negroponte/Guy Weinzapfel: „Architecture-by-Yourself. An Experiment with Computer Graphics for House Design“, in: *SIGGRAPH* 76 (1976), S. 74–78.

10 Yona Friedman: „About the Flatwriter“, in: *Pro Domo*, Barcelona 2006, S. 137.

generate a two-dimensional (planar) graph whose node could be manually repositioned by the user.

While the graph was being manipulated by the user in this way, computer routines were once again working in the background by evaluating the topological arrangement and providing the user with error messages when it identified potential conflicts for the future geometric layout of the rooms. At the time, the system was limited to two possible warnings: “No single level arrangement can be found for this set of links” and “No arrangement is possible without some enclosed space.” This meant that, due to the desired spatial relations, the three-dimensional realization of the arrangement had to play out across more than one level, or that the insertion of additional in-between spaces, such as a corridor, became necessary. According to the authors, in the case of an expansion of the system to include additional background routines, the future alignment of the rooms could also be reviewed with further factors in mind, such as the view or the incidence of light. Once satisfactory data entry was reached after manipulation of the graph by the user, that is, once the system had stopped generating error messages, then the third step could be initiated: the conversion of topological structure into a NURBS model, where the area of each individual bubble was precisely equivalent to the predefined wishes. In the final step, the celllike entity was meant to serve the user as a base for the manual design of the final configuration of the rooms. The hand-drafted contours on the screen were likewise machine-monitored in the background, with any significant deviations from the previously defined area requirements shown. Once this procedure was completely finished, the bubbles were deleted.

In the conclusion of the essay “Architecture-by-Yourself” (1976) by Nicholas Negroponte and Guy Weinzapfel, the authors labeled the concept that they had just introduced as a “worst-case exercise in man-machine interaction.”[9] Apparently, the Architecture Machine Group did not consider their objective of connecting human and machine in the architectural design process to have been successfully addressed through this experiment. For Yona Friedmann, the results of the experiment proved sobering as well: “My experiment with MIT made me understand a fundamental disadvantage of using computers for design: they are too fast. People have their own thinking speed and computer thinking is too fast, too intransparent for them.”[10]

Architecture is not the only discipline where people are confronted with complex problems and seek to find new, computer-supported paths. Software tools are being developed in

9 Nicholas Negroponte and Guy Weinzapfel, “Architecture-by-Yourself: An Experiment with Computer Graphics for House Design,” *SIGGRAPH* 76 (1976), pp. 74–78.

10 Yona Friedman, “About the Flatwriter,” in *Pro Domo* (Barcelona, 2006), p. 137.

various branches of science, such as in molecular biology, with an aim to successfully harness human intuition in solving complex combinatory problems. In places where even highly evolved computer algorithms fail, humans shine with their naturally born capabilities. One such tool is called FOLD.IT (fig. 2). This software—created by molecular biologist David Baker and computer scientist and game developer Zoran Popović—uses humans' preeminent capacity for spatial perception to achieve scientific ends: the mandate of the "player" is to fold protein chains as optimally as possible. A chaotic initial constellation must be converted into an extremely compact form using the available tools. While the player manipulates the protein via a pleasing graphic interface, its form is continually being analyzed by the program. Problem areas are color-coded, and the player receives verbal feedback from the program, which points out any problems that arise and even offers recommendations that can lead to an improvement of the situation and should not be disregarded in the process. Points are given for each achievement made. This is based on measurement operations that gauge the degree of optimization using certain criteria. Through this graphic and verbal feedback, the system enhances the senses of its users. The immediate visualization of the successes and failures resulting from each operative move hones intuition to facilitate the next correct move.

Playing FOLD.IT is fun. But like any "serious game," the software also has a more earnest background. In molecular biology attempts are being made to bind proteins that exhibit very particular properties, such as CO_2 or the HIV virus species. Whether this is successful is strongly determined by the state of folding, according to Baker. The player approaches reality in an isomorphic manner, which means that inferences may be drawn between the virtual copy and reality. Spatial constellations for which a player receives a high number of points simultaneously represent promising scientific hypotheses. Such proteins are then artificially cultivated and tested in a laboratory. Neither the player nor the computer can satisfactorily solve this complex task alone.

FOLD.IT offers exactly that which Robin M. Hogarth calls a "kind" learning environment in his book *Educating Intuition*. For the psychologist who scientifically explores the strengths and limits of human intuition, the structure of the learning environment is decisive. Hogarth asserts that a broad spectrum of different learning environments exists, ranging from "kind" to "wicked." Kind environments allow us to keenly feel the consequences of our behavior since they provide consistent and clear feedback. Through the consequences of our decisions we learn whether and to what extent the chosen behavior furthers or detracts from our goals.[11] By contrast, "wicked"

11 See Tilmann Betsch, "Wie beeinflussen Routinen das Entscheidungsverhalten," *Psychologische Rundschau* 56, no. 4 (2005), pp. 261–70, esp. p. 267.

Auftrag an den „Spieler" lautet, Proteinketten möglichst optimal zu falten. Eine chaotische Anfangskonstellation muss mithilfe der bereitgestellten Werkzeuge in ein möglichst kompaktes Gebilde umgewandelt werden. Während der Spieler das Protein über ein ansprechendes grafisches Interface manipuliert, wird dessen Gestalt vom Programm laufend analysiert. Problemstellen werden farbig gekennzeichnet und der Spieler erhält ein sprachliches Feedback durch das Programm, welches die auftretenden Probleme benennt und sogar Empfehlungen ausspricht, was zur Verbesserung der Situation getan werden kann und dabei nicht vergessen werden dürfte. Für jeden erzielten Erfolg gibt es Punkte. Dahinter stecken Messoperationen, welche den Grad der Optimiertheit anhand bestimmter Kriterien messen. Durch das grafische und verbale Feedback erweitert das System die Sinne seiner Benutzer. Die unmittelbare Visualisierung der Erfolge und Misserfolge bei jeder Handlung schärft die Intuition für die nächste richtige Handlung.

Räumliche Konstellationen, für die ein Spieler eine hohe Punktzahl erhält, stellen zugleich vielversprechende wissenschaftliche Hypothesen dar.

FOLD.IT zu spielen macht Spaß. Wie jedes „Serious Game" hat die Software aber auch einen ernsthaften Hintergrund. In der Molekularbiologie sucht man nach Proteinen mit ganz besonderen Fähigkeiten, wie etwa CO_2 oder HIV-Viren an sich zu binden. Ob dies klappt, hängt laut Baker entscheidend von der Faltung ab. Das Modell, in welchem der Spieler agiert, verhält sich isomorph zur Realität, d.h. zwischen virtuellem Abbild und Wirklichkeit sind Rückschlüsse möglich. Räumliche Konstellationen, für die ein Spieler eine hohe Punktzahl erhält, stellen zugleich vielversprechende wissenschaftliche Hypothesen dar. Solche Proteine werden daraufhin im Labor künstlich gezüchtet und getestet. Weder der Spieler noch der Computer könnten diese komplexe Aufgabe allein befriedigend lösen.

FOLD.IT bietet genau das, was Robin M. Hogarth in seinem Buch *Educating Intuition* als „freundliche" Lernumwelt bezeichnet. Für den Psychologen, der die Stärken und Grenzen der menschlichen Intuition wissenschaftlich untersuchte, ist in diesem Zusammenhang die Struktur der Lernumwelt entscheidend. Nach Hogarth existiert eine weite Bandbreite 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.[11] 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 Wechsel-

11 Vgl. Tilmann Betsch: „Wie beeinflussen Routinen das Entscheidungsverhalten", in: *Psychologische Rundschau* 56,4 (2005), S. 261–270, hier S. 267.

wirkung von Handlung und Erfahrung eintritt. Von etwas, das sich unserer Erkenntnis entzieht, können wir auch nichts lernen.

Die Arbeitsumgebung des architektonischen Entwerfers kann im Allgemeinen als eher „boshaft“ bezeichnet werden. Durch den Einsatz von CAD-Programmen und assoziativen Gebäudemodellen (BIM) lassen sich architektonische Gebäudeentwürfe mittlerweile zwar wesentlich komfortabler entwickeln als zuvor auf dem Reißbrett, Feedback über die Qualitäten eines Entwurfs geben diese Softwareumgebungen jedoch nicht. Wie die Proteinfaltungen bei FOLD.IT stellt auch jeder architektonische Entwurf eine Hypothese dar. Ob und wie weit eine Entwurfsvariante den vielfältigen Anforderungen an ein zukünftiges Gebäude entspricht, unterliegt während des gesamten Entwurfsprozesses allein der Urteilskraft des Entwerfers. Durch das Fehlen eines begleitenden Feedbacks wird dem Architekten nichts von der Multitasking-Aufgabe Entwerfen abgenommen. Ob ein Gebäudeentwurf alle gewünschten Parameter erfüllt, ist für den Entwerfer keineswegs offensichtlich. Keiner unserer fünf Sinne ist in der Lage, Merkmale wie Fluchtwegslängen oder Raumgrößen direkt wahrzunehmen. Für jeden zugehörigen Nachweis der Erfüllung solcher Eigenschaften sind viele Handgriffe und teilweise mühsame Hilfskonstruktionen notwendig, die mentale Anstrengung erfordern. Dies raubt dem Entwurfsprozess häufig den „Flow“ – jenen erstrebenswerten „Zustand der mühelosen Konzentration,“ der laut dem Psychologen Mihaly Csikszentmihalyi beispielsweise beim Spielen eintritt und „so tief ist, dass [Spielende] ihr Gefühl für die Zeit, für sich selbst und für ihre Probleme verlieren“.[12]

Ob ein Gebäudeentwurf alle gewünschten Parameter erfüllt, ist für den Entwerfer keineswegs offensichtlich. Keiner unserer fünf Sinne ist in der Lage, Merkmale wie Fluchtwegslängen oder Raumgrößen direkt wahrzunehmen.

Architecture Machines 2.0. Zwischen den Problemstellungen des Faltens von Proteinen in der Molekularbiologie und dem Entwerfen von Gebäuden in der Architektur sind Parallelen zu beobachten: In beiden Fällen geht es um Formfindung. Die einzelnen notwendigen Eigenschaften der gesuchten Form sind zwar bekannt, nicht aber die räumliche Konstellation, welche diese befriedigend erfüllt. Ebenso ähneln sich die Strategien zur Unterstützung dieses Prozesses durch den Computer bei YONA und FOLD.IT. In beiden Programmen wird dem menschlichen Entwerfer ein Expertensystem beiseite gestellt, welches dem User ein klares Feedback gibt und deutliche Handlungsempfehlungen ausspricht, welche aus einer zuvor eingegebenen Wissensbasis abgeleitet werden. Während sich FOLD.IT jedoch von Anfang an einer großen und stetig wachsenden Zahl von „Mitspielern“ er-

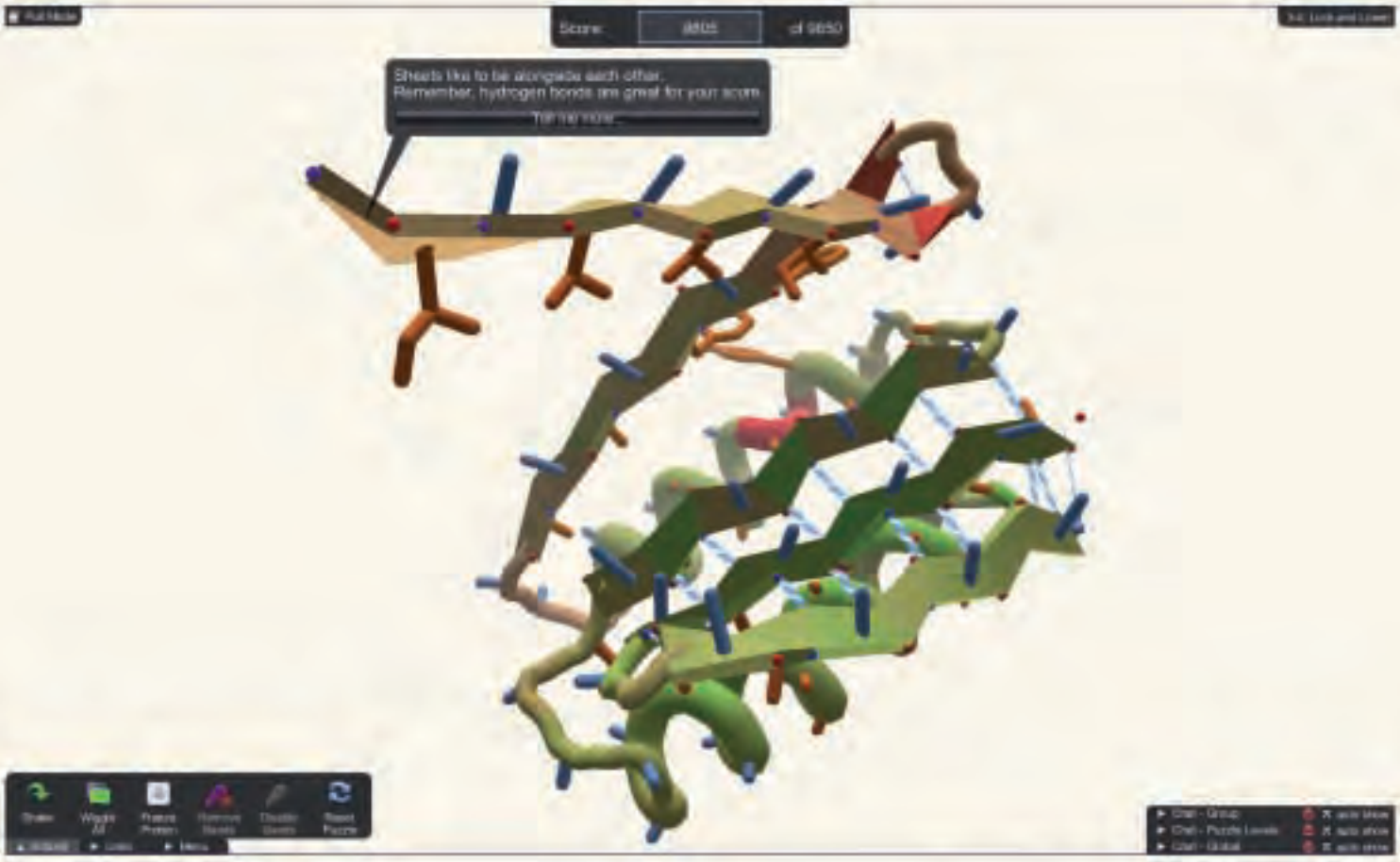
environments offer responses that are ambiguous, vague, too complex, temporally delayed, or sometimes even nonexistent. Hogarth notes that the latter environments impede adaptive learning, first allowing it to occur in a behavior-oriented way through the interplay between action and experience. Indeed, we cannot learn anything from something that eludes cognition.

The working environment of architectural designers may generally be considered quite “wicked.” Thanks to the implementation of CAD programs and associative building information modeling (BIM), designs for architectural structures are now much more conveniently developed than in previous times with a drawing board. However, these software environments do not provide feedback on the qualities of a design. As in the case of protein folding in FOLD.IT, each architectural design represents a hypothesis. Whether and to what extent a design variant meets the multiple demands of a future building is solely subject to the designer’s power of judgment during the entire design process. Without the accompanying feedback, an architect lacks support for his multitasking design work. In the eyes of the designer, whether a building design fulfills all parameters is not obvious by any stretch. Not one of our five senses is capable of directly perceiving features like room size or the length of emergency escape routes. In order to arrive at each required verification of such properties, many maneuvers and sometimes even laborious auxiliary structures are necessary, all of which entail mental strain. This often robs the design process of its “flow”—that “desirable state of effortless concentration” which, according to psychologist Mihaly Csikszentmihalyi, for instance arises during play and is so “intense that [those engaged in play] lose their sense of time, self, and their problems.”[12]

Architecture Machines 2.0. Parallels can be observed between the problem of folding proteins in molecular biology and the design of buildings in architecture: both cases deal with form generation. The individual properties necessary for the sought form have already been established, but the spatial constellation to satisfactorily do them justice has not. By the same token, the computer-based strategies for supporting this process in YONA and FOLD.IT are similar. In both programs the human designer is supported by an expert system which provides the user with clear feedback and makes distinct recommendations for action as derived from a knowledge base entered in the early stages. Yet while FOLD.IT has enjoyed a large and continually growing number of “teammates” from the outset, YONA was considered dissatisfactory early on. This discrepancy cannot be simply chalked up to the limited computing power of the devices available at the time. Nor did the limitations of

12 Mihaly Csikszentmihaly zitiert in Daniel Kahneman: *Schnelles Denken, langsames Denken*, München 2011, S. 56f.

12 Mihaly Csikszentmihalyi quoted in Daniel Kahneman, *Schnelles Denken, langsames Denken* (Munich, 2011), pp. 56–57.


Score:
Sheets like to be alongside each other.
Remember, hydrogen bonds are great for your score.

2

freute, galt YONA von Anfang an als gescheitert. Dies lässt sich nicht einfach durch die eingeschränkte Rechenleistung der damals zur Verfügung stehenden Computer erklären. Die Grenzen des Experiments der „Architecture Machine Group“ lagen wahrscheinlich auch nicht daran, dass der architektonische Formfindungsprozess einfach zu komplex ist, denn in dieser Hinsicht steht ihm die zellbiologische Aufgabe möglicherweise in nichts nach. Der Hauptgrund, warum das Projekt gescheitert war, lag vielleicht eher an der damals kursierenden Vision, die menschliche Kreativität könne zur Gänze durch eine Künstliche ersetzt werden. Friedman formulierte es klar und deutlich: „[…] the architect or the planner, who has for centuries kept his role as interpreter, has been eliminated from the process.“[13] Auch Negroponte äußerte sich zu diesem Vorhaben unmissverständlich: „The architects' primary function, […], will be served well and served best by computers.“[14] Bei Negroponte sollte sich der Architekt in der Universalität einer lernenden Maschine auflösen, bei Friedman durch die Partizipation der Bewohner.[15] Dieses absolute Ziel wurde mit Sicherheit verfehlt.

Die Maschine unterstützt den Menschen bei den räumlichen Verschachtelungen diskursiv durch visuelle und sprachliche Hinweise, greift aber niemals gestalterisch in diesen Prozess ein.

Der „Trick“ bei FOLD.IT, wie es David Baker formuliert, liegt gerade darin, „dafür zu sorgen, dass sich Mensch und Maschine mit ihren jeweiligen Stärken möglichst gut ergänzen.“[16] Keine Seite will die andere ersetzen. „Wir Menschen haben ein besonders gut ausgeprägtes räumliches Vorstellungsvermögen“, erklärt David Baker, „wir gucken uns das an und sagen: Aha, da ist ein Loch, und dieser Abschnitt könnte genau hineinpassen. Computer sehen das nicht.“[17] Die Maschine unterstützt den Menschen bei den räumlichen Verschachtelungen diskursiv durch visuelle und sprachliche Hinweise, greift aber niemals gestalterisch in diesen Prozess ein. Die Umformungen im Raum übernehmen bei FOLD.IT ausschließlich die Menschen, deren naturgegebene intuitive Fähigkeiten der Brute-Force[18] einer Maschine überlegen sind. Bei YONA sind diese Aufgaben noch nicht ganz so klar verteilt. Außerdem wurde die Komplexität, welche jeden architektonischen Entwurfsprozess auszeichnet, sehr vereinfacht und die einzelnen Handlungsschritte in eine streng lineare Abfolge gebracht – bis hin zu einem Punkt, an dem das System den Prozess nicht mehr umfassend abbilden konnte. Die Komplexität der Architektur ist jedoch irreduzibel. Die von YONA errechneten

13 Friedman: *Toward a Scientific Architecture*, S. 9 (wie Anm. 8).

14 Nicholas Negroponte: *Soft Architecture Machines*, Cambridge, MA 1975, S. 1.

15 Vgl. Vrachliotis: *Geregelte Verhältnisse*, S. 227 (wie Anm. 5).

16 David Baker zitiert in Dirk Asendorpf: „Die Nacht-Falter. Computerspiele helfen Forschern, komplexe Probleme zu lösen“, in: *Die Zeit*, 27.01.2011, Nr. 5, S. 39.

17 Ebd.

18 Die Brute-Force- oder Exhaustionsmethode ist ein Lösungsverfahren der Informatik, welches auf dem erschöpfenden Ausprobieren aller potentiellen Lösungen beruht.

the Architecture Machine Group experiment likely result from the architectural form-generation process merely being too complex, for in this respect the cytological problem is conceivably not inferior at all. The main reason why the project failed likely had more to do with the vision circulating at the time, namely, that of human creativity being entirely replaceable by its artificial counterpart. Friedman articulated it very clearly: "… the architect or the planner, who has for centuries kept his role as interpreter, has been eliminated from the process."[13] Negroponte, too, commented on this undertaking in no uncertain terms: "The architects' primary function … will be served well and served best by computers."[14] With Negroponte, the architect was to be dispelled by the universality of a learning machine, and with Friedman, by the participation of the residents.[15] This all-out target was surely missed.

In the case of FOLD.IT, the "trick," as David Baker phrases it, lies in "making sure that human and machine complement one another as well as possible in terms of their respective strengths."[16] Neither side wants to replace the other. "We humans have an exceptionally well developed capacity for spatial perception," explains Baker. "We look at something and say: Aha, that is a hole, and this section could fit in there perfectly. This is something that a computer wouldn't see."[17] The machine discursively supports the human when it comes to the interweaving of space by offering visual and verbal hints, but it never creatively intervenes in the process. With FOLD.IT, space is exclusively reshaped by humans since their natural intuitive abilities are superior to the brute force[18] of a machine. In the case of YONA, these tasks were not quite so clearly allocated. Moreover, the complexity inherent to each architectural design process was strongly simplified and the individual operative steps subjected to a strictly linear sequence—until reaching a point where the system could no longer comprehensively reflect the process. The complexity of architecture is indeed irreducible. In any case, the planning templates calculated by YONA required extensive postprocessing by humans. The developers of FOLD.IT, in turn, have been able to draw on four decades of research on artificial intelligence and complexity. Though the findings have occasionally been sobering, at the same time the awareness of the scientific world has been heightened for the

13 Friedman, *Toward a Scientific Architecture*, p. 9 (see note 8).

14 Nicholas Negroponte, *Soft Architecture Machines* (Cambridge, MA, 1975), p. 1.

15 See Vrachliotis, *Geregelte Verhältnisse*, p. 227 (see note 5).

16 David Baker quoted in Dirk Asendorpf, "Die Nacht-Falter: Computerspiele helfen Forschern, komplexe Probleme zu lösen," *Die Zeit*, January 27, 2011, no. 5, p. 39.

17 Ibid.

18 The "brute force" or "proof by exhaustion" method is a solution methodology used in computer science that is based on the exhaustive testing of all potential solutions.

astounding intuitive capabilities of humans (e.g., when applied to space), which to date have not been successfully imitated by technology. By contrast, the 1960s were characterized by a strong belief and high expectations in computer technology. The pioneering ideas of the Architecture Machine Group were radical and revolutionary. Our comparison with FOLD.IT has shown that some of the approaches developed back then still have topicality today. If taken further in a conceptual sense with a view to the current level of research and technology, today these approaches could serve to inspire a new generation of collaborative software tools for supporting the architectural design process. This process could also benefit from an intermeshing of human and machine, that is, from intuitive action and rational assessment methods. In this respect, the positive and playful approach taken by FOLD.IT may prove to be extremely useful.

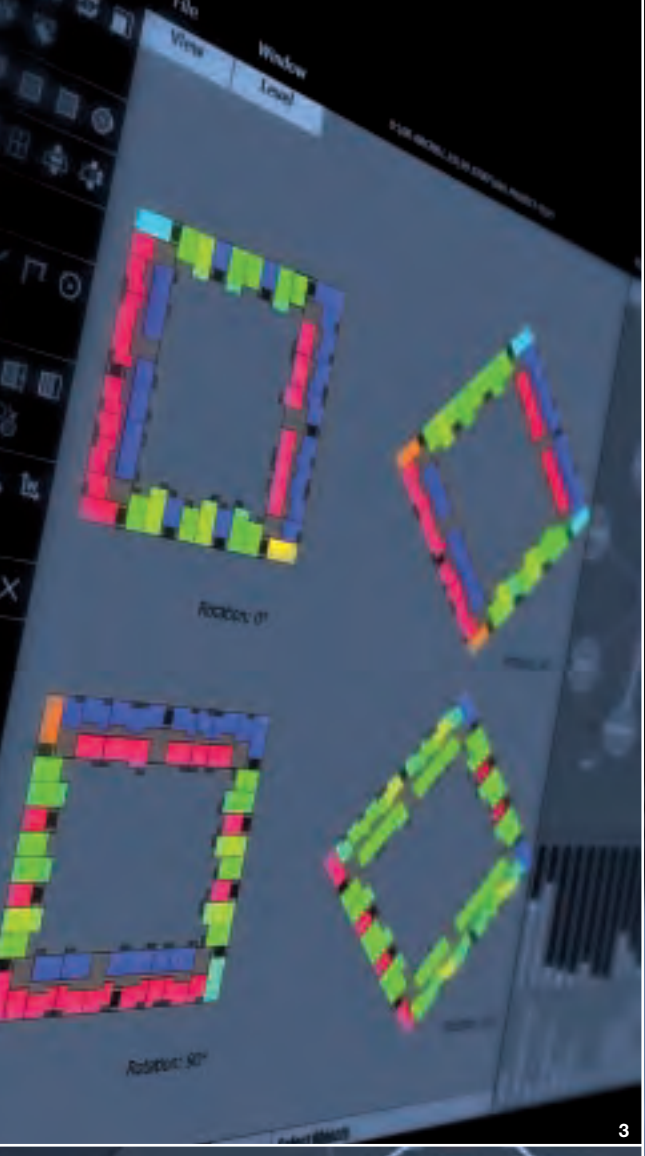
A concrete experiment is found in ARCHILL.ES, a custom-made software environment for architectural design conceptualized by the author. The rationale behind ARCHILL.ES is simple: the spatial development of a building should be manually carried out by the human designer, whose outstanding capacity for spatial perception and his inborn intuitive capabilities flow into the design. In carrying out this pursuit, he is provided with a "kind" environment where the potentials of the computer are utilized. While the designer manipulates the spatial arrangement via an easy-to-use graphic interface, in the background algorithms are running in parallel (so-called "architecture routines"). The latter monitor this process and constantly provide the designer with differentiated feedback on whether the many project objectives have already been met and to what extent. This information is given both visually and verbally.

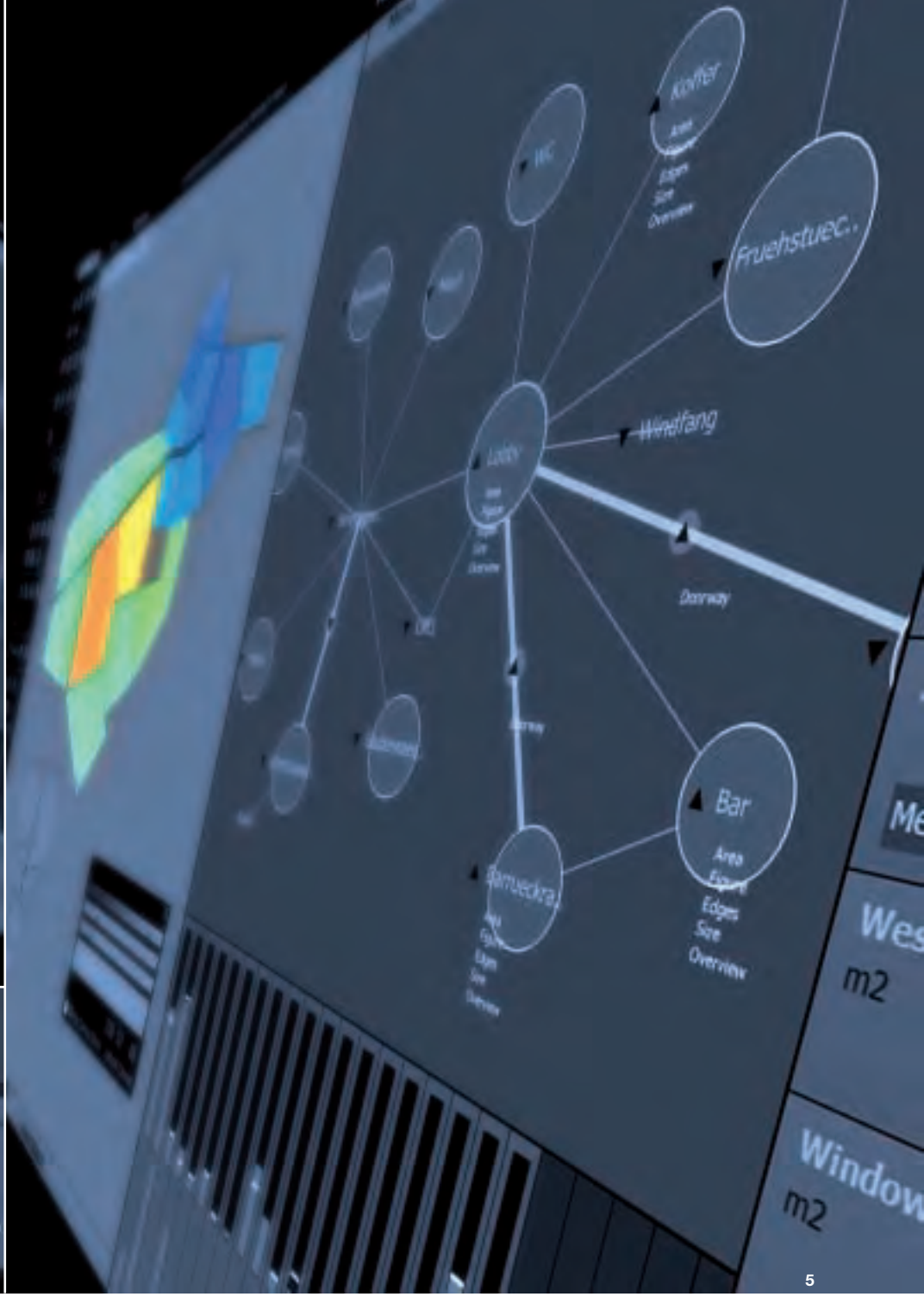
A wide range of design parameters can be machine-monitored in this way, such as those described in space- and function-related programs. But there is a hurdle to be overcome initially: still today design tasks are generally drafted in natural language. Here the specifications of individual rooms and their reciprocal relationships are usually expressed using terminology. The desire to create a "well-oriented" apartment or the claim that Room A should be situated "proximate" to Room B reflect seemingly clear planning measures from a human perspective. We associate the term "orientation" with a certain conceptual idea, as we do the term "proximity." Yet upon closer examination, even these terms prove to be exceedingly vague. If architectural design qualities are to be monitored by computers, then it is no longer enough to have only a rough idea of the meaning of the terms being employed. After all, it is necessary to supply the computer with an exact set of guidelines (i.e., an algorithm) which details how each of the features designated by the terms can be imposed. In order to convert the "pre-scripts" that are still today only readable by

Planungsvorlagen verlangten in jedem Fall nach einer umfassenden Nachbearbeitung durch den Menschen. Die Entwickler von FOLD.IT können auf vier Jahrzehnten der Erforschung von künstlicher Intelligenz und Komplexität zurückgreifen, deren Erkenntnisse zwar teilweise ernüchternd waren, zugleich aber auch die Aufmerksamkeit der Wissenschaft für die verblüffenden, technisch bis heute nicht imitierbaren, intuitiven Fähigkeiten des Menschen (beispielsweise im Umgang mit Raum) schärften. Die 1960er Jahre dagegen waren von einem starken Glauben und höchsten Erwartungen an die Computertechnologie geprägt. Die bahnbrechenden Ideen der „Architecture Machine Group" waren radikal und revolutionär. Der Vergleich mit FOLD.IT hat gezeigt, dass einige der damals entwickelten Ansätze noch heute Aktualität aufweisen. Weitergedacht und mit dem heutigen Stand der Forschung und Technik im Einklang können diese gegenwärtig einen inspirierenden Einfluss auf die Entwicklung einer neuartigen Generation kollaborativer Softwaretools zur Unterstützung des architektonischen Entwurfsprozesses ausüben. Auch der architektonische Gestaltungsprozess kann vom Ineinandergreifen von Mensch und Maschine, d.h., von intuitivem Handeln und rationalen Beurteilungsverfahren, profitieren. Der positive und spielerische Ansatz von FOLD.IT kann sich dabei als äußerst nützlich erweisen.

Einen konkreten Versuch stellt ARCHILL.ES, eine vom Autor konzipierte, für den architektonischen Entwurf maßgeschneiderte Softwareumgebung, dar. Das Grundprinzip von ARCHILL.ES ist einfach: Die räumliche Entwicklung eines Gebäudes soll manuell durch den menschlichen Entwerfer erfolgen, der sein herausragendes räumliches Vorstellungsvermögen und seine naturgegebenen intuitiven Fähigkeiten in den Entwurf einfließen lässt. Zur Ausführung dieser Tätigkeit wird ihm eine „freundliche" Umgebung geboten. Hierzu werden die Potenziale des Computers genutzt: Während der Entwerfer das räumliche Arrangement über ein einfach zu bedienendes grafisches Interface manipuliert, arbeiten im Hintergrund parallel dazu Algorithmen (sog. Architektur-Routinen), welche diesen Vorgang überwachen und dem Entwerfer laufend ein differenziertes Feedback darüber geben, ob und wie weit die zahlreichen Zielvorstellungen bereits erfüllt sind. Die Hinweise erfolgen sowohl visuell als auch sprachlich.

Eine Vielzahl von Entwurfsparametern, wie sie beispielsweise in Raum- und Funktionsprogrammen beschrieben werden, können auf diese Weise maschinell überwacht werden. Dazu muss zunächst aber eine Hürde überwunden werden: Entwurfsaufgaben werden bis heute üblicherweise in natürlicher Sprache verfasst. Die Spezifikationen der einzelnen Räume und deren Beziehungen untereinander erfolgen dabei in der Regel durch Begriffe. Der Wunsch nach einer „gut orientierten" Wohnung oder die Forderung, dass Raum A „nahe" 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 selbst diese Begriffe jedoch äußerst vage. 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 schließlich eine exakte Handlungsanleitung (i.e., ein Algorithmus) bereitgestellt werden, die beschreibt, wie jedes der durch die Begriffe bezeichneten Merkmale erhoben werden kann. Um die bis heute nur von Menschen lesbaren Vor-Schriften in von Computern ausführbare Programme umzuwandeln, ist es nötig,





3 Entwurfstool Design tool ARCHILL.ES: Analyse einer Wohnanlage Analysis of a housing complex, 2013 © Martin Emmerer
4 Entwurfstool Design tool ARCHILL.ES: Parameter-Setup am Mischpult Parameter setup at the mixing desk, 2013 © Martin Emmerer
5 Entwurfstool Design tool ARCHILL.ES: Topologie einer Entwurfsaufgabe Topology of a design job, 2013 © Martin Emmerer

humans into programs run by computers, it is necessary to pay attention to the relationship between terms and objects. A machine-based counterpart needs to be posited vis-à-vis the mysterious "associative bond"[19] that allows us to recognize the qualities of a building design.

To this end, the author developed the concept of *algorithmic explication*. First it is necessary to translate words that in everyday language are often used in a vague manner into unequivocal terms. When applying this method, which is commonly used in empirical science, a vague term undergoes a semantic analysis in that as many of its linguistic indicators are listed as possible. In the case of "well-oriented" living space, a simplified process might play out as follows: "a room with one or more windows, which is at least partially oriented to a southern or western direction." Here it is possible, but not necessarily imperative, to take into account already existing scientific studies of the term. It is surely possible to define "well-oriented" in a different or more elaborate way, but the semantic analysis of terms is less concerned with finding the "only correct" method, than with reaching an intersubjective agreement on as plausible of a definition as possible. After completing a thorough semantic analysis, it is time to operationalize the term by furnishing particulars about how the indicators of the term in question, as listed in the analytical definition (semantic analysis), can be specified concrete terms. In our example of "well-oriented" space, this would entail verifying whether this particular room has at least one window with an axis that deviates less than 90 degrees from the geographic south or west direction. The previously vague idea of the concept of "well-oriented" living space has now been turned into a measureable and calculable specification. Implementing this process in algorithm form to address the property "orientation" in a suitable digital building model is now no longer a real challenge. After completing these steps, a computer, for example, would be in a position to identify all "well-oriented" apartments in a sophisticated housing complex in a matter of seconds. By repeatedly carrying out this technique to address all terms that might be enlisted on an everyday basis to describe design tasks, a vocabulary is built which is not only associated with intersubjective concepts, but which also fosters a dialogue with the machine. Such an explicit vocabulary is used in the computer program ARCHILL.ES (figs. 3–5).

In this environment the design process is configured as follows: the user starts out by modeling the topological structure of the design task in graph form, with the nodes and edges being "weighted" with various desired properties (fig. 5). This is carried out by a docking of "property modules." The significance of the spatial and relational properties is explicitly explained in a brief text, which contains not only a terminological definition

19 Ferdinand de Saussure, *Course in General Linguistics* (1959; repr., New York, 2011), p. 66.

der Beziehung zwischen Begriffen und Objekten Aufmerksamkeit zu schenken. Dem geheimnisvollen „Band der Assoziation",[19] welches es uns ermöglicht, Qualitäten eines Gebäudeentwurfes zu erkennen, wäre ein maschinelles Pendant gegenüberzusetzen.

Dazu wurde vom Autor das Konzept der *Algorithmischen Explikation* entwickelt. Zunächst ist es notwendig, die häufig unscharfen Begriffe der Alltagssprache in deutliche Begriffe umzuwandeln. Bei diesem in den empirischen Wissenschaften gängigen Vorgang wird ein vager Begriff einer Bedeutungsanalyse unterzogen, indem möglichst alle seiner sprachlichen Indikatoren aufgezählt werden. Im Falle eines „gut orientierten" Wohnraums könnte das vereinfacht etwa so aussehen: „Ein Raum mit mindestens einem Fenster, welches zumindest anteilig in Richtung Süden oder Westen orientiert ist." Eine Berücksichtigung bereits vorliegender wissenschaftlicher Auseinandersetzungen mit dem Begriff ist dabei möglich, aber nicht zwingend. Es ist sicherlich möglich, „gut orientiert" anders oder ausführlicher zu definieren. Bei der Bedeutungsanalyse von Begriffen geht es aber weniger darum, die „einzig richtige" Methode zu finden, sondern vielmehr um die intersubjektive Vereinbarung einer möglichst plausiblen Definition. Nach einer vollständigen Bedeutungsanalyse kann die Operationalisierung des Begriffes erfolgen, indem Angaben gemacht werden, wie die in der analytischen Definition (Bedeutungsanalyse) aufgezählten Indikatoren des betrachteten Begriffs konkret erfasst werden können. In unserem Beispiel des „gut orientierten" Raumes würde dies bedeuten, zu überprüfen, ob der betreffende Raum mindestens ein Fenster aufweist, dessen Achse um weniger als 90° von der geografischen Süd- bzw. Westrichtung abweicht. Die zuvor vage Vorstellung des Begriffs von „gut orientierten" Wohnräumen ist nun in eine mess- und berechenbare Angabe umgewandelt worden. Die Implementierung dieses Vorgangs in einen Algorithmus zur Erfassung des Merkmals „Orientierung" in einem geeigneten digitalen Gebäudemodell stellt jetzt keine große Schwierigkeit mehr dar. Danach ist ein Computer beispielsweise in der Lage, in Sekundenbruchteilen alle „gut orientierten" Wohnungen einer komplexen Wohnanlage zu identifizieren. Durch die wiederholte Anwendung dieses Verfahrens 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. Ein solches explizites Vokabular wird vom Computerprogramm ARCHILL.ES genutzt (Abb. 3–5).

Das differenzierte Setup der Variablen erfolgt wie über ein spezielles Mischpult, auf dem die gewünschte Ausprägung jedes Merkmals vom Benutzer über Schieberegler eingestellt wird.

Der Entwurfsprozess gestaltet sich in dieser Umgebung wie folgt: Der Benutzer modelliert zunächst die topologische Struktur der Entwurfsaufgabe

19 Ferdinand de Saussure: *Grundfragen der allgemeinen Sprachwissenschaft*, Berlin 2001, S. 77.

als Graph, dessen Knoten und Kanten durch diverse Wunschmerkmale „gewichtet“ werden (Abb. 5). 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 wie über ein spezielles Mischpult, auf dem die gewünschte Ausprägung jedes Merkmals vom Benutzer über Schieberegler eingestellt wird (Abb. 4). 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. Wenn das Setup vollständig ist, kann mit dem räumlichen Entwurf begonnen werden. Die Konturen der gewünschten Räume werden gezeichnet und auf einer oder mehreren Ebenen angeordnet. Ebenso werden wichtige architektonische Elemente wie Türen, Fenster, vertikale Erschließungselemente etc. platziert. Hinter dem sehr einfach zu bedienenden Zeichenprogramm steckt ein radikal vereinfachtes *Building Information Model*, gerade detailliert genug, um den vergleichsweise niederen Detailgrad eines üblichen Wettbewerbsbeitrags abbilden zu können. Die klare Trennung von Topologie und Geometrie ist entscheidend. Der räumliche Entwurf und die vorweg modellierte topologische Struktur der Entwurfsaufgabe bestehen parallel nebeneinander. Die Knotenpunkte des Graphen werden über die Raumbezeichnungen mit ihren geometrischen Repräsentanten im Entwurf synchronisiert und die Abweichungen von den vorab definierten Zielvorstellungen (durch sog. Architektur-Routinen) berechnet. Die Ergebnisse werden durch Farben dargestellt. Abbildung 3 zeigt vier Varianten derselben Wohnanlage. Für jede Wohneinheit wurde der Status „gut orientiert“ definiert. Der Plan ist genordet. Durch die Rotation des Gesamtbaukörpers um jeweils 45% verändert sich die Orientierung der einzelnen Wohneinheiten. Die Farbnuancen geben dem Entwerfer Auskunft über den Grad der Erfüllung des zuvor formulierten Zieles. Dabei steht Rot für „sehr gute“, Grün für „gute“ und Blau für eher „schlechte“ Orientierung. Die farbige Visualisierung stellt ein eindeutiges und unmittelbares Feedback dar und erlaubt es dem Entwerfer, intuitiv und planerisch gezielt auf mögliche Schwachpunkte seines Entwurfes zu reagieren bzw. die beste Variante auszuwählen.

Durch die Verwendung expliziter Begriffe bei der Beschreibung von Entwurfsparametern werden klare Vereinbarungen getroffen. Die Maschine übernimmt in der Folge die Funktion eines Schiedsrichters.

Die Software stellt bereits ein Basisrepertoire implementierter Merkmale zur Modellierung von Entwurfsaufgaben zur Verfügung. Dieses soll laufend erweitert werden. Als Umschlagplatz der Merkmalsmodule wird die

(semiotic analysis) but also explanations of how and on which level of measurement the property is covered. The differentiated setup of variables ensues as if using a special mixing console with which the user adjusts the desired manifestation of each property via slide controls (fig. 4). The scales of qualitative properties are made up of terminological categories, while the slide controls for quantitative properties represent numerical intervals. While for instance the property module “orientation” only permits a choice between “very good,” “good,” and “poor,” the module “room size” allows the desired number of square meters to be stipulated at will by sliding the controls. When the setup is completed, the spatial design can begin. The contours of the desired rooms are drawn and arranged across one or more levels. Moreover, important architectural elements are placed, like doors, windows, vertical access elements, et cetera. Operating in the background of this very easy-to-use drawing program is a radically simplified building information model that is just detailed enough to be able to portray the comparatively low degree of detail needed for a typical competition proposal. The clear separation between topology and geometry is decisive. Spatial design and the topological structure of the design task modeled in advance exist in parallel. The nodal points in the graph are synchronized with their geometric representatives in the design using the room specifications, and deviations from the previously defined objectives (through the so-called “architecture routines”) are calculated. The results are presented using different colors. Figure 3 shows four variants of the same housing complex. For each housing unit the status “well-oriented” has been defined. The plan is geared toward the north. In rotating the entire building structure by 45 percent respectively, the orientation of the individual housing units is shifted. Shades of color provide the designer with information about the degree to which the preformulated goal has been fulfilled. Here red stands for “very good,” green for “good,” and blue for a rather “poor” orientation. The colorful visualization represents clear and immediate feedback and makes it possible for the designer to take an intuitive and planning-related approach to targeting potential weak points in his design and to thus react or select the best variant.

The software already places at one’s disposal a basic repertoire of implemented properties for modeling design tasks. The aim is to continually expand this repertoire. The platform WWW.ARCHILL.ES is designed to serve as a reloading point for the property modules. Newly developed modules can be downloaded like apps for testing. The plan for the future is to make it possible for ARCHILL.ES users to implement their own property modules, which can then be loaded by the system as plug-ins. This will enable the vocabulary of the system to expand much more quickly and thus allow design tasks to be modeled in an increasingly differentiated way. The know-how of experts from different specialty fields could automati-

cally flow into the design process. This would allow a use of the software in a teaching context to assume a highly experimental and playful character. The fact that the tool is also designed to handle the modeling of differentiated suitability functions of building designs equally opens up new possibilities in the research of generative design strategies. Modules allowing the computer to actively intervene in form-generation processes are also conceivable here.

To a certain extent, designing in such an environment is akin to a game. Like on any gameboard, here, too, "an absolute and peculiar order reigns."[20] Although the rules of the game are clearly defined, they are set by the slider on the mixing console. By using explicit terms in describing design parameters, clear agreements are reached. The machine subsequently takes on the role of referee. If a room is colored blue, for example, then it is "poorly oriented." Whoever wants to keep this up for debate is going against the laws inherent to the game. While the "player" tries to arrive at red color fields and achieve high scores, serious architecure is created step by step. "The noblest games, such as chess, are those that organize a combinatory system of places in a pure *spatium* infinitely deeper than the real extension of the chessboard and the imaginary extension of each piece."[21] Just as constructivism saw architecture as the "mother of all arts," after the digital turn the act of designing buildings might now have a chance at becoming the "noblest of all games." And the architect, or homo faber, would become the homo ludens.[22] So be house DJs and accept the support of machines. Develop routines and "play" with new instruments—or, to paraphrase Beastie Boy Mike D., "let the computer rock!"[23]

Translation Dawn Michelle d'Atri

Plattform WWW.ARCHILL.ES dienen. Neu entwickelte Merkmalsmodule können hier wie Apps heruntergeladen und getestet werden. Es ist geplant, den Benutzern von ARCHILL.ES in Zukunft auch die Implementierung eigener Merkmalsmodule zu ermöglichen, welche dann als Plug-ins in das System geladen werden können. Auf diese Weise kann sich das Vokabular des Systems rascher erweitern und Entwurfsaufgaben können dadurch immer differenzierter modelliert werden. Experten unterschiedlicher Fachrichtungen können ihr Know-how automatisiert in den Entwurfsprozess einfließen lassen. Dadurch kann die Anwendung auch im Bereich der Lehre sehr experimentellen und spielerischen Charakter annehmen. Da es sich bei dem Tool gleichzeitig um ein Werkzeug zur Modellierung differenzierter Fitnessfunktionen von Gebäudeentwürfen handelt, eröffnet dies ebenso neue Möglichkeiten im Bereich der Erforschung generativer Entwurfsstrategien. Dabei sind auch Module zum aktiven Eingriff des Computers in den Formfindungsprozess denkbar.

In gewisser Weise gleicht das Entwerfen in einer solchen Umgebung einem Spiel. Wie auf jedem Spielbrett herrscht auch hier eine „eigene und unbedingte Ordnung".[20] Die Spielregeln sind klar definiert, sie werden über die Slider am Mischpult aufgestellt. Durch die Verwendung expliziter Begriffe bei der Beschreibung von Entwurfsparametern werden klare Vereinbarungen getroffen. Die Maschine übernimmt in der Folge die Funktion eines Schiedsrichters. Färbt sich ein Raum blau, ist er zum Beispiel „schlecht orientiert". Wer darüber noch diskutieren will, bricht die inneren Gesetze des Spiels. Während der „Spieler" roten Farbflächen und High-Scores nacheifert, entsteht Schritt für Schritt ernsthafte Architektur. „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."[21] 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, wird zum homo ludens.[22] Seid Haus DJs und nehmt die Hilfe der Maschine an. Entwickelt Routinen und „spielt" auf neuen Intrumenten – oder in den Worten des Beastie Boy Mike D: „Let the Computer rock!"[23]

20 See Johan Huizinga, *Homo Ludens: A Study of the Play Element in Culture* (London, 1955), p. 10.

21 Charles J. Stivale, "Appendix: Deleuze's 'How Do We Recognize Structuralism?'," *The Two-Fold Thought of Deleuze and Guattari: Intersections and Animations* (New York, 1998), p. 264.

22 Huizinga, *Homo Ludens*, p. ix (see note 20).

23 Diamond Michael aka "Mike D.," cited in Thomas Hüetlin, "Drei Jungs im Müll," *Der Spiegel*, August 1, 1994, p. 140.

20 Vgl. Johan Huizinga: *Homo Ludens. Vom Ursprung der Kultur im Spiel*, Hamburg 2011, S. 19.

21 Gilles Deleuze: *Woran erkennt man den Strukturalismus*, Berlin 1992, S. 19.

22 Vgl. Huizinga: *Homo Ludens*, S. 1 (wie Anm. 20).

23 Diamond Michael aka „Mike D." zitiert nach: Thomas Hüetlin: „Drei Jungs im Müll", in: *Der Spiegel*, 1.8.1994, S. 140.

GAM.

ARCHITECTURE MAGAZINE **10**

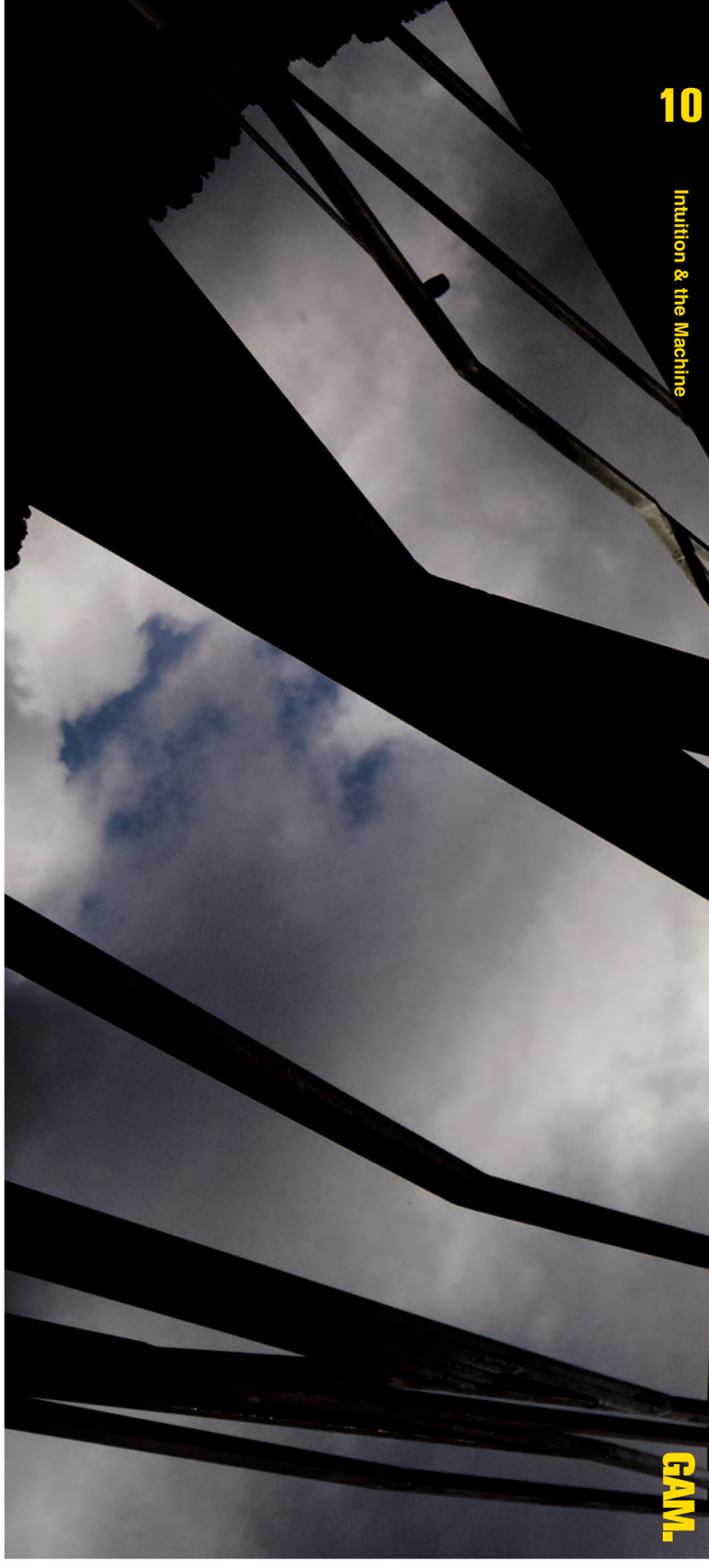
Digitale Medien bereichern die Architekturproduktion auf vielfältige Weise: Sie schaffen eigene Simulations-, Entwurfs- und Planungsinstrumente, neuartige Fabrikations- und Konstruktionsprozesse und generieren neue Raumerfahrungen. Digitale Maschinen sind längst keine Maschinen im herkömmlichen Sinn mehr, denn sie erweitern unsere Sinneswahrnehmung und nehmen uns immer mehr kognitive Tätigkeiten ab. **GAM.10 – „Intuition & the Machine“** diskutiert die Auswirkungen dieser Entwicklung für intuitiv gesteuerte Prozesse in der Architektur. Entsteht hier eine Konkurrenz für unsere menschlichen Fähigkeiten? Oder eröffnen sich neue Potenziale für einen sinnlicheren, intuitiveren Umgang mit Komplexität im Architekturentwurf? **GAM.10** stellt architektonische Projekte, Konzepte und Entwurfsexperimente vor, die aus der Interaktion von digitalen, architektonischen und sinnlichen Erfahrungswelten hervorgegangen sind.

Digital media are enriching architectural production in many ways: through simulation, design and planning instruments, new fabrication and construction processes, and the generation of new spatial experiences. Digital machines have long ceased to be machines in the traditional sense. They are capable of expanding our sensual perception and taking on more and more of our cognitive tasks. **GAM.10 – “Intuition & the Machine”** discusses the consequences of these developments for intuitively steered processes in architecture. Are our very own human capacities being challenged? Or are new potentials opening up for a more sense-sensitive, intuitive handling of complexity in architectural design. **GAM.10** presents architectural projects, concepts, and design experiments that have resulted from the interaction of digital, architectural, and sensory worlds of experience.

Mit Beiträgen von With contributions by

Edith Ackermann
Nathalie Bredella
Mario Carpo
Kristof Crolla
Benjamin Dillenburger
Kathrin Dörfler
Martin Emmerer
Fabio Gramazio
Sean Hanna
Michael Hansmeyer
Urs Hirschberg
Ludger Hovestadt
Matthias Kohler
George L. Legendre
Ammar Mirjan
Felix Raspall
Romana Rust
François Roche
Skylar Tibbits
Tobias Wallisser
Makoto Sei Watanabe
Jan Willmann
Chris Younès

Cover: Shin Minamata MON generated by KeiRiki program
Architekt/Foto Architect/photo: MAKOTO SEI WATANABE



ISBN 978-3-99043-620-2
9 783990 436202
ambra-verlag.at