



ŠTRUKTÚRA GENEROVANÁ
NA ZÁKLADE PRINCÍPOV
RASTU URBÁNNYCH CELKOV-
ILUSTRÁČNÝ OBRÁZOK

A STRUCTURE GENERATED ON
THE BASIS OF PRINCIPLES OF THE
GROWTH OF URBAN UNITS-
ILLUSTRATIVE VISUALIZATION

Autor Author: research / Ján Pernecký,
2018

Nová argumentácia za digitálnu paradigmu

New Argumentation for the Digital Paradigm

Ján Pernecký, Tomáš Tholt

We consider the conceptual digital branch of architecture to be a direct continuation of the well-articulated discourses of the 20th century and before. In our view, it has the capacity to pose fundamentally critical questions of the discipline of architecture beyond its own paradigm. Before we attempt to shift the current architectural paradigm, we need to argue yet again for the conceptual novelty and the potential of the digital design to become a relevant moment in the history of architecture.

Digitally processed tasks require neither a notation understandable to the human mind nor static data but instead dynamic algorithms. We have conducted experimental research into the tools for the design and fabrication, and of their generative capacities, which gives a possibility of creating the form during its fabrication.

The elementary level of the digital design is automation – the use of computers for the autonomous processing of repetitive tasks with no differentiation between the repetitions. The input parameters per element differ in the parametric design paradigm, creating series of similar non-identical elements (in)formed by internal or external influences. The highest level of the digital design so far – the emergent or generative approach – simulates a bottom-up complex system of a multitude of agents equipped with an internal logic. Its result is the morphogenesis of an emergent object featuring novel properties beyond the sum of its parts. Inspiration by nature is not merely mimetic (mimicking outer forms) but truly morphogenetic – utilizing similar forces and processes as those found in nature. The proposed “Design by Behavior” implements architectural intentions into the agents of complex system simulations, resulting in the

expectation of strictly architectural emergent outcomes to allow for new objects and processes, which will address the current and further discourse of architecture.

The conventional design process expects the finding of an ultimate solution for a single task, whereas the parametric design first creates an abstract design tool offering a multitude of viable solutions. Thus the true object of creation is the abstract promise of an object with all its potential embodiments. In architectural practice, there is an inherent pressure to concretize the abstraction and anchor it in the real Euclidian and civilizational world.

The latest period dealing with context was post-modernism, when the architects had the courage to create tensions and complicated relations with a historical, architectural or formal context (existing or new). The currents since the late 1990's have merely adjusted themselves to, kept back from, emphasized the existing context or completely ignored it. Digital design offers the possibility to create non-trivial contexts and implicitly react to them – not through post-modern intuition and wit but instead through complex, non-random ways in various scales of space, time, syntactic and semantic dimensions. In consequence, we should consider the process and the object of creation not only as a complex system of carefully designed architectural agents, but also as a stated machine with a non-deterministic context. The architectural and procedural elements will then be implemented while reacting implicitly to the context. Once we are educated in digital design, we have a good chance to create semantic exuberance and emergent relations beyond cultural semiotics and technological optimization.

V roku 2009 sme identifikovali dve tendencie, ktoré mali ambíciu stať sa hlavným prúdom architektúry: pragmatickú tvorbu s participatívnou inklúziou používateľa a digitálny dizajn. Kým tá prvá aj dnes stále ašpiruje na dominanciu, digitálny dizajn si už zrejme našiel svoju pozíciu pri optimalizácii technologicko-ekologických a funkcionalisticko-prevádzkových aspektov inak konvenčne navrhutej architektúry. Stal sa tak najmä pomocnou disciplínou iných prístupov k architektonickej tvorbe. Súčasný návrhový proces v architektonickej praxi možno označiť za tradičný, so stále výraznejším prenikaním pragmatických optimalizačných a simulačných nástrojov na riešenie parciálnych a jasne ohraničených úloh. Ešte pred desiatimi rokmi pritom bolo možné vnímať technicko-optimalizačný a koncepčne digitálny smer ako rovnako silné.

Tú druhú tendenciu považujeme za priame pokračovanie ucelených prúdov architektúry 20. storočia a staršej. Veríme, že dokáže otvárať fundamentálne, kritické otázky disciplíny architektúry aj vzniku jednotlivého diela nad úrovňou svojej špecifickej paradigmy. Pri ostatných spomenutých prístupoch nedokážeme identifikovať kapacitu tvoriť nové alebo autonómne diela, a tým pádom adresovať témy nad rámec „tu a teraz“.

Nositelmi inovácie návrhových a fabrikačných konceptov sú interdisciplinárne akademické tímy. Do praxe presakuje ich činnosť vo forme umeleckých inštalácií, pavilónov a dizajnových objektov menších mierok. Zdá sa však, že na hlavný diskurz v disciplíne architektúry má ich

výskum len malý vplyv, paradoxne však tvorí významné ideové východiská pre iné odvetvia ľudskej činnosti.¹

Koncepty digitálneho dizajnu, čo by mohli zmeniť architektúru ako takú, sú v úzadí azda viac než v minulosti. Súčasná štruktúra spoločnosti sa pritom mení vplyvom nových ekonomických modelov (decentralizácia, zdieľaná ekonomika), spoločenských zmien (sociálne siete a bubliny, virtuálny priestor) a technológií (umelá inteligencia, on-demand-production, blockchain). Architektúra, aby si ako disciplína zachovala relevanciu, by mala tieto tendencie reflektovať vo svojich témach, v procesoch aj v samotnom móde existencie. Preto sa dnes javí, že pred snahou o posun architektonickej paradigmy treba opäť lobiť za konceptuálnu novosť a s tým spojený potenciál digitálneho dizajnu stať sa relevantným momentom v histórii architektúry.

Digitálna fabrikácia a materiálová inteligencia

Digitálne navrhovanie, evaluácia a fabrikácia, ktoré sa dnes stávajú priemyselným štandardom, prebiehajú strojovo, a preto podklady na ktorúkoľvek z týchto činností vyžadujú inú notáciu, než je konvenčná projektová dokumentácia. Notáciou digitálne spracúvaných činností nemusia byť ľuďmi priamo čitateľné zápisy ani statické dáta – môžu to byť algoritmy, teda sekvencie príkazov, podmienených vetvení a cyklov, čo sa vykonávajú počas procesu návrhu či realizácie, reagujúc na vnútorné alebo vonkajšie okolnosti.

V ostatných rokoch sme sa zamerali na experimentálny výskum návrhových a fabrikačných nástrojov aj ich generatívnych kapacít – možnosti tvoriť počas procesu výroby. Potreba dobre poznať materiál a nástroj, ktorý ho obrába, tak významne posúva pozíciu architekta a náplň jeho práce prvýkrát od Albertiho oddelenia projektovej a stavebnej činnosti.²

Výrobný nástroj chápeme ako súbor vzťahov, príkazov a úkonov vedúcich k výsledku. Do procesu digitálnej výroby a návrhu vstupujú vlastnosti materiálu, ktorý (na rozdiel od materiálu používaného v masovej produkcii) môže byť priestorovo alebo časovo heterogénny, prinášajúc výsledky, čo nemožno simulovať, pričom sa s každým opakovaním rovnakého procesu získajú rôzne výsledky. Fabrikačný nástroj s implementovanou rozhodovacou logikou reaguje flexibilne na vlastnosti materiálu alebo prostredia. V iteráciách nasledujúcich po sebe aktualizuje vstupné dáta algoritmu, a preto je takto generovaný objekt determinovaný interakciou algoritmu, nástroja aj materiálu.³ Digitálne (CAM) fabrikačné technológie umožňujú masovo vyrábať variabilitu bez toho, aby bola výroba časovo alebo materiálovo náročnejšia.⁴ Dnes je teda možné za masovú prispôsobenosť (*mass customization*) argumentovať ekonomickou a sociálnou výhodnosťou, rovnako ako sa v čase priemyselnej revolúcie argumentovalo za masovú výrobu (*mass production*).

Na platforme *rese arch* sme zorganizovali viacero experimentálnych workshopov zameraných na materiálovú inteligenciu a robotickú fabrikáciu. Na workshope s Mateuszom Zwierzyckim⁵ sme skúmali 3D aditívnu tlač plastového filamentu na predpätú textilnú plochu s dvojitém zakrivením a následné samovoľné (de)formácie takto vzniknutých objektov. Počas workshopu s Adrianom Krężlikom a Kacperom Radziszewskim⁶ sa skúmali vlastnosti heterogénneho taveného a rezaného polystyrénu a miera predvídania spávania roboticky obrábaného materiálu. Stuard Maggs na svojom workshope skúmal recyklovateľnú 3D tlač z vosku a Christoph Müller⁸ roboticky fabrikoval neunifikované nafukovacie stavebné bloky.

Komplexné systémy, bottom-up rozhodovanie, emergencia

V teoretických textoch sme pomenovali a opísali tri existujúce stupne digitálneho dizajnu,⁹ ktoré možno aplikovať na proces návrhu a fabrikácie architektonického diela. Základným stupňom je automatizácia, keď sa počítače používajú na autonómne spracovanie repetitívnych úloh bez zmeny medzi jednotlivými opakovaniami. Pri parametrickom návrhu sa jednotlivé repetície od seba odlišujú zmenou parametrov, a vznikajú tak série podobných, nie však identických prvkov (in)formovaných vnútornými alebo vonkajšími parametrami. Aktuálne najvyšším stupňom je emergentný alebo generatívny dizajn, pri ktorom veľké množstvo činiteľov (agentov) s vlastnou vnútornou logikou spolupôsobí pri morfogénéze (vzniku tvaru) objektu. Agenti často zastupujú nezlučiteľné idey a pri vzájomnej interakcii sa ich sily sčítavajú alebo neutralizujú. Sústava takýchto agentov v prostredí s iteratívnym vývojom (v nadväzujúcich opakovaníach) sa nazýva komplexný systém. Ten je v neustálej nerovnováhe, osciluje medzi hraničnými stavmi, reagujúc na aktuálnu situáciu. Fungovanie takéhoto bottom-up systému možno prirovnať k ekosystémom, ktoré sa evolučne vyvíjajú v meniacich sa okrajových podmienkach. Biologická evolúcia pritom nehľadá optimálne

riešenie konkrétnej úlohy, je len neustálou reakciou na meniaci sa kontext. Dôsledkom komplexného systému sú emergentné javy – vznik situácií, ktoré nie sú iba súčtom javov jednotlivých agentov, ale majú kvalitatívne nové vlastnosti. Emergencia je jav, ktorý vzniká, keď nedokážeme lineárnou logikou predpovedať správanie systému na základe vlastností častí, z ktorých sa skladá.¹⁰

Interakcie medzi agentmi sú deterministické (majú jasné príčiny), ale nelineárne (nemožno ich predvídať analyticky), a preto aj emergentné efekty takéhoto návrhového procesu sú neočakávané, ale nenáhodné. Emergentné javy vznikajú na rôznych úrovniach a medzi úrovňami od molekulárnej až po ekosystémy a sociálne pohyby.¹¹ V súčasnosti možno úspešne simulovať správanie skupín živočíchov, rastlín, prvkov neživej prírody, buniek alebo neurónov a interpretovať ich do (proto) architektonických objektov. Inšpirácia prírodou tu nie je mimetická (napodobňujúca vonkajšie znaky), ale skutočne morfogenetická – teda uplatňujúca podobné tvorivé sily a procesy, aké existujú v prírode.

„Je ťažké uveriť, že niečo nerozmyšľajúce a také mechanické ako algoritmus by mohlo vytvoriť také úžasné veci. Nezáleží na tom, aký pôsobivý je produkt algoritmu, proces v jeho pozadí nie je ničím iným ako súborom nezávislých po sebe nasledujúcich krokov, bez inteligentného riadenia; sú automatické svojou podstatou, pracovným procesom stroja.“¹²

Hľadanie dizajnu správáním

A práve implementáciu javov pozorovaných v prírode možno považovať za slabinu existujúceho generatívneho dizajnu. Nami navrhovaný „Dizajn správáním“ spočíva v implementácii architektonických tém do komplexných systémov, pričom očakávame, že takto vzniknuté emergentné javy budú striktné z architektonickej domény a práve takto môžu vzniknúť objekty a procesy, ktoré budú adresovať súčasný a budúci diskurz architektúry ako disciplíny. Podobnú úvahu rozvíjal text *Parametrizácia architektonických teórií*,¹³ ktorý sa neskôr stal prvým manifestom platformy *rese arch*. Na túto tému sme zorganizovali konferencie,¹⁴ kde sme sa snažili identifikovať, či táto téma rezonuje vo svetovom diskurze.

Abstrakcia a produkcia sérií

Pri hľadaní myšlienkových východísk digitálneho dizajnu má zmysel obracať sa na filozofiu, prírodné vedy, spoločenské štúdie, kognitívnu vedu, no dôležité je poznať aj počítačovú vedu (zaujímavá by bola úvaha, či práve programovanie nie je remeslom digitálnej architektúry a počítačová veda jeho metodikou).

Algoritmus je dobre definovaná sekvencia funkcií a príkazov, ktoré majú práve jednu interpretáciu. Slúži na riešenie určitého typu úlohy (teda nie jednu konkrétnu úlohu), a preto je abstrakciou všetkých takýchto úloh, ktorú pre jednotlivé inštancie algoritmu konkretizujú vstupné parametre. Algoritmom je napríklad matematická funkcia f s parametrami a , b : vráť výsledok $a + b$. Pre parametre 2 a 3 budeme funkciu volať (vyhodnocovať) zápisom do kontajnera c vlož výsledok f pre konkrétne parametre 2, 3. Kontajner s názvom c má teraz hodnotu 5. Algoritmus musí byť abstraktný, preto funkcia g bez parametrov: vráť výsledok $2 + 3$ nie je algoritmom, pretože je riešením práve jednej konkrétnej úlohy, a nie typu úlohy.

Už samotná podmienka abstraktnosti je dobrým príkladom ontologického rozdielu medzi konvenčným a digitálnym (tu parametrickým) navrhovaním. Kým pri konvenčnom návrhovom procese autor hľadá jedno konkrétne riešenie úlohy, pri parametrickom navrhovaní vzniká najprv abstraktný návrhový nástroj (algoritmus), ponúkajúci väčšie množstvo riešení (vyplývajúcich z rôznych vstupných parametrov), z ktorých autor alebo iný abstraktný stroj vyberá jedno najlepšie.¹⁵ Zároveň sú však všetky ponúkané riešenia validne, a preto existuje akási abstraktná vízia (fikcia, prísľub) diela, ktorá nadobúda všetky formy, čo môžu vzniknúť z povolených rozsahov vstupných parametrov a na výstupe spĺňajú kriteriálnu evaluáciu. Pre tvorbu digitálnych architektov je typická produkcia sérií diel. Nie preto, lebo výpočtová sila algoritmov umožňuje tvorbu redundantného množstva objektov. Skutočným dielom je práve abstraktná vízia objektu so všetkými jeho formami a všetky konkretizované inštancie tejto vízie sú rovnocenné. V architektonickej tvorbe existuje inherentný tlak túto abstrakciu konkretizovať a ukotviť v reálnom euklidovskom a civilizačnom priestore. Digitálna tvorba architektúry sa teda od programovania líši v tom, že jej výsledkom nie je iba funkčný algoritmus – stroj na plnenie určitých úloh –, ale aj jeho výstupy. Na druhej strane, od konvenčnej architektonickej tvorby sa zasa odlišuje tým, že súčasťou diela je aj proces jeho vzniku – algoritmus.

Opozícia k explicitnému pragmatizmu

Existujú mnohé, vzájomne kombinovateľné alebo exkluzívne paradigmy a štýly vývoja softvéru – programovania. My sa zameriame na implicitný a explicitný štýl.

Pri použití explicitného (deklaratívneho) štýlu sú všetky komponenty algoritmu vždy opísané všetkými svojimi parametrami, a to aj vtedy, keď je tento opis redundantný. Pri implicitnom štýle sú parametre adoptované z kontextu – z takzvaných stavov – a vytvárame tak tzv. stavový stroj (*finite state machine*).¹⁶ Rozdiel medzi explicitným a implicitným programovacím štýlom demonštrujeme na jednoduchšej grafickej úlohe. V prípade explicitného štýlu by vykreslenie jedného kruhu so stredom v súradnici 10, 10, polomerom 5, obrysovou farbou červenou a výplňovou farbou modrou vyzeralo takto (A1):

1 / vykresli kruh pre konkrétne parametre 10, 10, 5, červená, modrá

Ukážme si zložitejší príklad (B1):

1 / vykresli kruh pre konkrétne parametre 10, 10, 10, červená, modrá

2 / vykresli kruh pre konkrétne parametre 10, 10, 7, červená, modrá

3 / vykresli kruh pre konkrétne parametre 10, 10, 5, žltá, modrá

4 / vykresli kruh pre konkrétne parametre 20, 20, 5, žltá, zelená

Takto vzniknú tri sústredné kruhy so stredom na súradnici 10, 10: prvý s polomerom 10, červeným obrysom a modrou výplňou, druhý s polomerom 7, červeným obrysom a modrou výplňou, tretí s polomerom 5, žltým obrysom a modrou výplňou. Štvrtý kruh bude mať stred na súradnici 20, 20, polomer 5, žltý obrys a zelenú výplň.

Predstavme si algoritmy s rovnakou funkcionalitou napísané implicitným štýlom a implementovaním stavového stroja.

Prvý príklad (A2):

1 / presuň sa o 10, 10

2 / nastav mierku na 5-násobok súčasnej mierky

3 / nastav obrysovú farbu na červenú

4 / nastav výplňovú farbu na modrú

5 / vykresli kruh

Druhý príklad (B2):

1 / presuň sa o 10, 10

2 / nastav mierku na 10-násobok súčasnej mierky

3 / nastav obrysovú farbu na červenú

4 / nastav výplňovú farbu na modrú

5 / vykresli kruh

6 / nastav mierku na 7/10-násobok súčasnej mierky

7 / vykresli kruh

8 / nastav mierku na 5/7-násobok súčasnej mierky

9 / nastav obrysovú farbu na žltú

10 / vykresli kruh

11 / presuň sa o 10/5, 10/5

12 / nastav výplňovú farbu na zelenú

13 / vykresli kruh

Algoritmus napísaný v explicitnom štýle je absolútne deterministický – každý príkaz vykonáva za každých okolností presne jednu vec. Tá je vďaka explicitnému zápisu dobre čitateľná z pohľadu na samotný zápis príkazu bez ohľadu na to, v akom kontexte sa nachádza.

Každý riadok zápisu v implicitnom štýle stavového stroja buď mení globálny kontext (B2: 1, 2, 3, 4, 6, 8, 9, 11, 12), alebo naň reaguje (B2: 5, 7, 10, 13). Znamená to, že kontext stavového stroja prepíname do rôznych stavov (počiatok súradnicového systému, mierka, farba) a potom voláme funkciu (procedúru), ktorá na tento stav reaguje. V našom príklade je to volanie vykresli kruh, ktoré je v riadkoch 5, 7, 10, 13 totožné, rozdielny je však kontext. Zo samotného pohľadu na riadok, ktorý vykresľuje kruh, nevieme zistiť, aký kruh sa skutočne vykreslí. Dôležité sú riadky, čo mu predchádzajú, tie, ktoré menia kontext. Niektoré z týchto prepínačov stavov sú deterministické (B2: 3, 4, 9, 12) – z pohľadu na ne možno predvídať, aký stav bude nastavený, pretože nastavujú konkrétnu hodnotu ohľadu na to, v akom stave sa kontext práve nachádza. Iné príkazy sú nedeterministické (B2: 1, 2, 6, 8, 11), teda reagujú na kontext a menia ho relatívne k jeho aktuálnemu stavu. Preto aj výstupy takýchto prepínačov stavov závisia od kontextu a menia sa podľa pozície (v kóde).

Tieto príklady sú však lineárne, a preto veľmi jednoduché. Algoritmy často obsahujú bifurkácie – rozvetvenia na dve možné vetvy kódu, iteratívne cykly – opakovania tej istej časti kódu s postupne sa meniacimi parametrami a externé vstupy. Kým explicitný kód sa bude vždy správať rovnako bez ohľadu na predchádzajúce časti kódu (pri výhradnom použití konštant a tzv. immutable – nemenných premenných), výstupy implicitného nedeterministického kódu za takýchto okolností nemožno predvídať. Ak by sme na začiatok implicitných nedeterministických príkladov A2 alebo B2 pridali riadok meniaci kontext, každý z nasledujúcich riadkov (okrem deterministických B2: 3, 4, 9, 12) by mal za dôsledok úplne iné zmeny stavov (B2: 1, 2, 6, 8, 11) alebo výstupy (B2: 5, 7, 10, 13). Inými slovami, explicitný štýl sa vyhýba vedľajším efektom (*side effects*), implicitný je na ich použitie založený.

Logiku explicitných, implicitných či deterministických alebo nedeterministických prístupov môžeme pozorovať aj v reálnom svete – najmä na príkladoch zariadení vyrobených človekom. Kolískový prepínač je deterministický a explicitný, schodiskový vypínač alebo tlačidlový prepínač je nedeterministický a implicitný, domáce spotrebiče, ktoré najprv nastavíme a potom spustíme (práčka, mikrovlnná rúra), sú deterministické a implicitné, vodovodný kohútik je nedeterministický a explicitný, páková batéria je deterministická a implicitná. V prírode a v spoločnosti sa však stretávame najmä s nedeterministickými a implicitnými javmi – javy, mierky, objekty sa vzťahujú na iné referenčné javy a pri ich zmene sa menia aj javy na ne relačne naviazané. To samo je dobrou motiváciou na jeho skúmanie v autorskej tvorbe.

V architektúre identifikujeme komunikáciu a prácu s kontextom naposledy v období postmoderny, keď mali autori odvahu vytvárať napäté a zložité vzťahy s historickým, architektonickým či formálnym kontextom (existujúcim alebo vlastným). Prúdy konca 20. a začiatku 21. storočia sa kontextu najmä prispôbujú, ustupujú mu, umocňujú ho alebo úplne ignorujú. Práve digitálnym návrhom a fabrikáciou možno vytvárať netriviálne kontexty a implicitne na ne reagovať, už nie postmodernou intuíciou a vtipom, ale ich dôslednou analýzou a najmä ich vytváraním v rôznych mierkach a doménach – rozmerových, časových, relačných, významových.

Vnímajme preto dielo a proces jeho vzniku nielen ako komplexný systém s mnohými agentmi, ktorých správanie navrhne tak, aby zastupovalo architektonické témy (teda implementujeme Dizajn správaním), ale aj ako stavový stroj s nedeterministicky zakódovanými stavmi a architektonické aj procedurálne prvky implementujeme implicitne do takto vytvoreného kontextu. Poučení digitálnym dizajnom máme dobrý predpoklad na to, aby sme vytvorili obsahovú bohatosť a relačné vzťahy – teda emergentné javy – nad rámec semiotiky z kultúrneho hľadiska a nad rámec analytickej optimalizácie z technickej stránky. Čo je potom v prípade architektúry kontext stavového stroja a čo prvky od neho závislé? Je možné, aby boli jedným aj druhým? Aký je vzájomný vzťah prvkov implicitne závislých od kontextu a prvkov explicitných (od kontextu nezávislých) a je možné navrhnúť objekt výhradne implicitný a nedeterministický? Aké nové kvality sa objavia a možno ich čítať rovnakou metodikou ako vznikli?

ING. ARCH. MGR. ART.
JÁN PERNECKÝ
RESE ARCH

janper@rese-arch.org

ING. ARCH. TOMÁŠ THOLT
RESE ARCH

FAKULTA ARCHITEKTÚRY STU

Námestie slobody 19
812 45 Bratislava
Slovensko

tomas@rese-arch.org

1 CARPO, Mario, 2011. *The Alphabet and the Algorithm*. Boston: The MIT press, s. 16 – 48.

2 CARPO, Mario, 2017. *The second digital turn*. Boston: The MIT press, s. 2 – 5.

3 SCHUMACHER, Patrik, 2008. *Parametricism – A New Global Style for Architecture and Urban Design* [online] [cit. 21. 4. 2018]. Dostupné z: <http://www.patrikschumacher.com/Texts/Parametricism%20-%20A%20New%20Global%20Style%20for%20Architecture%20and%20Urban%20Design.html>

4 CARPO, Mario, 2011. *The Alphabet and the Algorithm*. Boston: The MIT press, s. 7.

5 ZWIERZYCKI, Mateusz, 2015. Workshop: *rese arch LAB*. Fellows: Ján Pernecký, Štefan Lopušný, Sofia Zourelli, Alexander Ahmad, Jiří Vítek, Ralf

Bliem, Konrad Zellner, Tomáš Tholt a Löffler Robert. Bratislava.

6 KŘEŽLIK, Adrian a RADZISZEWSKI, Kacper, 2015. Workshop: *rese arch LAB 2*. Fellows: Štefan Lopušný, Sofia Zourelli, Alexander Ahmad, Andrei Padure, Ralf Bliem, Konrad Zellner, Tomáš Tholt, Ján Pernecký a Bartolomiej Bieńkowski. Bratislava.

7 MAGGS, Stuart, 2015. Workshop: *rese arch LAB 3*. Fellows: Ralf Bliem, Giorgio Castellano, Sebastian Comanescu, Shima Moradi, Andrei Raducanu, Konrad Zaremba a Konrad Zellner. Bratislava.

8 MÜLLER, Christoph, 2017. Workshop *rese arch LAB 4*. Fellows: Christian Kolbeck, Stefan Pirintchev, Konrad Zellner, Anel Bucan, Michael P. Schultes. Bratislava.

9 PERNECKÝ, Ján, 2013. *Digitálna architektúra* [online], s. 2 – 4. [cit. 21. 4. 2018] Dostupné z: <http://www.scribd.com/doc/245809676/romboid-digital-na-architektura>

10 PAVLOVSKÝ, Martin, 2004. *Emergencia*. [online] [cit. 21. 4. 2018] Dostupné z: <http://neuron.tuke.sk/tino/emergence.html>

11 DE LANDA, Manuel, 2011. *Philosophy and Simulation*. Londýn: Continuum International Publishing Group.

12 „It is hard to believe that something as mindless and mechanical as an algorithm could produce such wonderful things. No matter how impressive the products of an algorithm, the underlying process always consists of nothing but a set of individually mindless steps succeeding each other without help of intelligent

supervision; they are ‚automatic‘ by definition; the workings of an automaton.“ DENNETT, Daniel, 1996. *Darwins Dangerous idea: evolution and the meanings of life*. New York: Simon and Schuster, s. 59.

13 PERNECKÝ, Ján, 2009. *Parametrizácia architektonických teórií*. Archív autora.

14 rese arch, Design by data, UMPRUM A3, Volumes Paris; konferencie: *rese arch MEETUP Bratislava 2015, Paris 2016, Prague 2017, Paris 2018*.

15 UHRÍK, Martin, 2010. *Digitálna architektúra*. Bratislava: Eurostav, s. 51 – 66.

16 SHEAD, Mark, 2011. *State Machines – Basics of Computer Science*. [online] [cit. 21. 4. 2018] Dostupné z: <https://blog.markshead.com/869/state-machines-computer-science/>