Designing (tools (for designing (tools (for ...))))

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

Thomas Fischer
Dr. phil. (University of Kassel)

Volume I

School of Architecture and Design
Faculty of the Constructed Environment
RMIT University
February 2008
I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Thomas Fischer
Hong Kong, 26th of February 2008
During the course of this study, the following related papers have been published:


I gratefully acknowledge the valuable support and feedback I have received from my supervisors

Mark Burry  
*Royal Melbourne Institute of Technology University, Australia*  
and  
John Frazer  
*Queensland University of Technology, Australia*

and the indispensable and generous mentoring I have received throughout this research from

Ranulph Glanville  
*CybernEthics Research, United Kingdom*

as well as the feedback, the suggestions and a lot of proofreading from

Timothy Jachna  
*The Hong Kong Polytechnic University, Hong Kong*
I also wish to thank all panelists, besides those mentioned above, who have commented on, criticised and challenged my research during the RMIT Graduate Research Conference presentations including (in alphabetical order):

Andrew Burrow (RMIT), Peter Downton (RMIT), Pia Ednie-Brown (RMIT), Zeynep Mennan (Middle East Technical University), Ted Krueger (Renssealer Polytechnic Institute and RMIT), Thomas Kvan (University of Melbourne), Mark Goulthorpe (deCOi and MIT), Jeff Malpas (University of Tasmania), Michael Ostwald (University of Newcastle and RMIT), Julieanna Preston (Massey University), Nicholas Ray (University of Cambridge and Nicholas Ray Associates), Dennis Shelden (Gehry Technologies), Mark Taylor (Victoria University of Wellington and RMIT), Juliette Peers (RMIT), Mike Xie (RMIT)

For their generous support and feedback I also wish to thank (in alphabetical order):

Simon Anson (ARUP London), Chris Bosse (PTW, Sydney), Kenneth Brakke (Susquehanna University), Stuart Bull (ARUP Sydney), Tristram Carfrae (ARUP Sydney), Cristiano Ceccato (Gehry Partners), Mao-Lin Chiu (National Cheng Kung University, Tainan), Hans Dehlinger (Kassel University), Steven Downing (ARUP Sydney), Torben Fischer (Mahr Fertigungsmesstechnik, Göttingen), Ernst Hafen (University of Zürich), Christiane M. Herr (The University of Hong Kong) Richard Hough (ARUP Sydney), A. Scott Howe (University of Hong Kong), Taysheng Jeng (National Cheng Kung University, Tainan), Klaus Knebel (MERO), Alexander Koutamanis (Delft University of Technology), Lin Hsuan-Cheng, (National Cheng Kung University, Tainan), Luo Chien Rung, (National Cheng Kung University, Tainan), Rüdiger Lutz (ARUP Berlin), Andrew Maher (RMIT), Anitra Nelson (RMIT), Paul Nicholas (ARUP Melbourne and RMIT University), Reinhard Schmidek (MERO), Jaime Sanchez-Alvarez (MERO), Nicole Schadewitz (The Open University, UK), Fabian Scheurer (designproduction and ETH Zürich) Shen Yang-Ting, (National Cheng Kung University, Tainan), Siegfried Stolz (MERO), Alvise Simondetti (ARUP London), Michael Orion (Phoenix, Arizona), Hugh Whitehead (Foster and Partners), Peter Wood (University of Wellington), Robert Woodbury (Simon Fraser University), Margaret Woods (RMIT), Gerard de Zeeuw (Universiteit van Amsterdam), Gregor Zimmermann (Kassel University).

I also wish to thank everybody at the Spatial Information Architecture Lab and the postgraduate community at the School of Architecture and Design at RMIT for welcoming me into a challenging and stimulating research environment.
# CONTENTS

Volume I

<table>
<thead>
<tr>
<th>Declaration</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Published work</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>Summary</td>
<td>xvi</td>
</tr>
</tbody>
</table>

0 Background

| 0.1 Introduction                | 1 |
| 0.2 A personal perspective      | 2 |
| 0.3 Scope and limitations of this thesis | 6 |
| 0.4 Structure of this thesis    | 9 |
| 0.5 Terms used in this thesis   | 11 |

1 Review of literature and work of others

| 1.1 Introduction                | 2 |
| 1.2 Tool use, tool making and computer-aided design | 27 |
| 1.2.1 Tool use and tool making in humans and other animals | 27 |
| 1.2.2 Engineering design        | 30 |
| 1.2.3 Software engineering and computer programming | 31 |
| 1.2.4 Design methods and design research | 36 |
| 1.2.5 Design systems and CAAD research | 42 |
| 1.2.6 Tools and media           | 46 |
| 1.3 Generative design and inspiration from Nature | 48 |
| 1.3.1 Biological analogies in science and design | 48 |
| 1.3.2 Parametric tools for designing | 49 |
| 1.3.3 Evolutionary tools for designing | 52 |
| 1.3.4 Cellular automata tools for designing | 55 |
| 1.3.5 Evolutionary Developmental Biology | 58 |
| 1.3.6 Limitations of biological analogies | 60 |
## 1.4 Space grid geometry

1.4.1 Polyhedra and soap bubble geometry ........................................... 62
1.4.2 Design of architectural space frame structures ................................. 69
1.4.3 Architectural geometry rationalisation ............................................. 76
1.4.4 Pre-, post- and co-rationalisation .................................................. 84

## 1.5 Models of knowledge growth in science and design ............................. 87

1.5.1 From objective truth to discursive growth of knowledge ....................... 88
1.5.2 The relationship between design and science .................................... 92
1.5.3 Lack of methods of enquiry ............................................................ 95
1.5.4 Second-order cybernetics .............................................................. 98

## 1.6 Summary and research questions ..................................................... 101

## 2 Research strategies ........................................................................... 105

2.1 Introduction ......................................................................................... 106
2.2 Toolmaking strategies ........................................................................ 108
  2.2.1 Cellular development ..................................................................... 111
  2.2.2 Teaching scripting ......................................................................... 113
  2.2.3 Subway construction ....................................................................... 113
  2.2.4 Parametric cell geometry ............................................................... 115
  2.2.5 Re-orientation towards foam-like grids ......................................... 116
  2.2.6 Self-supporting façade system ...................................................... 117
  2.2.7 Development of irregular space grids ............................................ 118
  2.2.8 Exploring co-rationalisation ......................................................... 119
  2.2.9 Tsunami Memorial Design Competition ........................................ 121
  2.2.10 Postgraduate design workshop ...................................................... 122
  2.3 Reflective strategies .......................................................................... 123
  2.4 Summary ............................................................................................ 125

## 3 Toolmaking study .............................................................................. 126

3.1 Introduction ......................................................................................... 127
3.2 Cellular development ........................................................................... 128
3.3 Teaching scripting ............................................................................... 134
3.4 Subway construction ............................................................................ 136
3.5 Parametric cell geometry .................................................................... 138
3.6 Re-orientation towards foam-like grids .............................................. 145
3.7 Self-supporting façade system ............................................................. 154
3.8 Development of irregular space grids ................................................... 158
3.9 Exploring co-rationalisation ............................................................... 169
3.10 Tsunami Memorial Design Competition ............................................ 186
3.11 Postgraduate design workshop ............................................................ 191
3.12 Summary ............................................................................................ 204

## 4 Reflective analysis .............................................................................. 208

4.1 Introduction ......................................................................................... 209
4.2 Aspects of reflection ........................................................................... 209
  4.2.1 Conversation and novelty generation ............................................ 211
<table>
<thead>
<tr>
<th>Volume I</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Translation of a packed sphere into radial and circumferential vectors</td>
</tr>
<tr>
<td>0.2</td>
<td>Grids of rhombic dodecahedra, Kelvin-like and Weaire-Phelan-like polyhedra</td>
</tr>
<tr>
<td>1.1</td>
<td>Galapagos finch probing for insects</td>
</tr>
<tr>
<td>1.2</td>
<td>Carpenter’s workshop</td>
</tr>
<tr>
<td>1.3</td>
<td>Dimensions of design space accessibility</td>
</tr>
<tr>
<td>1.4</td>
<td>Skulls of <em>Hyrachyus agarius</em> and <em>Aceratherium tridactylum</em></td>
</tr>
<tr>
<td>1.5</td>
<td>Paramorphosis from neutral box to articulated Paramorph</td>
</tr>
<tr>
<td>1.6</td>
<td><em>Swiss Re</em> building in London</td>
</tr>
<tr>
<td>1.7</td>
<td>Prototype of a generative design tool for wine glasses</td>
</tr>
<tr>
<td>1.8</td>
<td>Evolutionary structural optimisation of the <em>Passion Façade</em></td>
</tr>
<tr>
<td>1.9</td>
<td>Early three-dimensional cellular automata structure</td>
</tr>
<tr>
<td>1.10</td>
<td>Cellular automata as generators of space grid systems</td>
</tr>
<tr>
<td>1.11</td>
<td>Cellular automata as generators of space grid systems</td>
</tr>
<tr>
<td>1.12</td>
<td>Cubic wire frame with soap films and “Kelvin’s bedspring”</td>
</tr>
<tr>
<td>1.13</td>
<td>Rhombo-dodecahedral shape of plant cells and octet truss</td>
</tr>
<tr>
<td>1.14</td>
<td><em>Reptiles</em> enclosure designed by John Frazer</td>
</tr>
<tr>
<td>1.15</td>
<td>Visually irregular truss structures</td>
</tr>
<tr>
<td>1.16</td>
<td>Atrium structure of <em>Federation Square</em></td>
</tr>
<tr>
<td>1.17</td>
<td>Gaudi’s hanging model</td>
</tr>
<tr>
<td>1.18</td>
<td>Xenakis’ sketch development for the rationalisation of the <em>Philips Pavilion</em></td>
</tr>
<tr>
<td>1.19</td>
<td>Initial, final schemes for the <em>Sydney Opera House</em>, sketches and sphere model</td>
</tr>
<tr>
<td>1.20</td>
<td><em>Beijing National Swimming Centre</em> space frame</td>
</tr>
<tr>
<td>1.21</td>
<td><em>Beijing National Swimming Centre</em> grid structure development</td>
</tr>
<tr>
<td>1.22</td>
<td>Boulders derived from Danzer tiling</td>
</tr>
<tr>
<td>1.23</td>
<td>Geometric principle for translation surfaces.</td>
</tr>
<tr>
<td>1.24</td>
<td>Serendipity in scientific progress</td>
</tr>
<tr>
<td>1.25</td>
<td>Smalley’s paper model of C_{60}</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>2.1 Structure of the toolmaking study and the reflective analysis</td>
<td>106</td>
</tr>
<tr>
<td>2.2 Truss rationalisation method</td>
<td>121</td>
</tr>
<tr>
<td>3.1 Enclosure structure based on Frazer’s <em>Reptiles</em> design</td>
<td>131</td>
</tr>
<tr>
<td>3.2 Fractal structures generated using <em>Zellkalkül</em></td>
<td>131</td>
</tr>
<tr>
<td>3.3 Sample page of student notes</td>
<td>135</td>
</tr>
<tr>
<td>3.4 Simulation of cellular subway station and tunnel development</td>
<td>137</td>
</tr>
<tr>
<td>3.5 Close-packing of cells of variable size</td>
<td>141</td>
</tr>
<tr>
<td>3.6 Cell geometry, triangulation and vertex ranges, STL illustration</td>
<td>144</td>
</tr>
<tr>
<td>3.7 Tetrahedral and octahedral movement ranges of two vertex types</td>
<td>144</td>
</tr>
<tr>
<td>3.8 Detail of parametrically altered close-packed cells</td>
<td>144</td>
</tr>
<tr>
<td>3.9 “Watercube” as displayed in winning competition entry</td>
<td>146</td>
</tr>
<tr>
<td>3.10 Close-packed rhombic dodecahedra and spheres</td>
<td>149</td>
</tr>
<tr>
<td>3.11 Wooden model showing break-down of rhombic pyramids</td>
<td>149</td>
</tr>
<tr>
<td>3.12 Decomposition of dodecahedra into self-similar rhombic pyramids</td>
<td>150</td>
</tr>
<tr>
<td>3.13 Decomposition and variation in (local) density</td>
<td>150</td>
</tr>
<tr>
<td>3.14 Search space for alternative midpoints, shifted midpoint</td>
<td>151</td>
</tr>
<tr>
<td>3.15 Moved midpoint, variable density, removal of dodecahedral frame</td>
<td>152</td>
</tr>
<tr>
<td>3.16 <em>Parasol</em> shapes irregular tetrahedra and node connection detail</td>
<td>155</td>
</tr>
<tr>
<td>3.17 Construction of flat wall from irregular tetrahedra</td>
<td>156</td>
</tr>
<tr>
<td>3.18 Dodecahedra composed of open sheet triangles and spherical node units</td>
<td>157</td>
</tr>
<tr>
<td>3.19 Three-dimensional Voronoi diagram inscribed in a cube</td>
<td>159</td>
</tr>
<tr>
<td>3.20 Processing of Voronoi structure in <em>Surface Evolver</em></td>
<td>160</td>
</tr>
<tr>
<td>3.21 Bottom-up constructed lattices</td>
<td>161</td>
</tr>
<tr>
<td>3.22 Adjustment of slight mismatch in lattice cluster</td>
<td>162</td>
</tr>
<tr>
<td>3.23 Visual irregularity in different space grids and from varying perspectives</td>
<td>165</td>
</tr>
<tr>
<td>3.24 Stick model of irregular liquid foam</td>
<td>166</td>
</tr>
<tr>
<td>3.25 Packing of truncated octahedra and of Weaire-Phelan-like cells</td>
<td>166</td>
</tr>
<tr>
<td>3.26 My bedspring: patch, stressed node, unstressed node</td>
<td>168</td>
</tr>
<tr>
<td>3.27 Bubble distances evolved to comply with distance rule</td>
<td>171</td>
</tr>
<tr>
<td>3.28 Interdependence of vertex locations</td>
<td>171</td>
</tr>
<tr>
<td>3.29 Two different mechanical calculators</td>
<td>174</td>
</tr>
<tr>
<td>3.30 Regular two-dimensional lattice of 100 cells evolved with different “pressures”</td>
<td>177</td>
</tr>
<tr>
<td>3.31 Elevations of two straight walls, both of a single layer of bubbles</td>
<td>178</td>
</tr>
<tr>
<td>3.32 Clustering of strut lengths</td>
<td>178</td>
</tr>
<tr>
<td>3.33 Struts with threaded bolt joints</td>
<td>179</td>
</tr>
<tr>
<td>3.34 Exterior surface smoothing in <em>Surface Evolver</em></td>
<td>180</td>
</tr>
<tr>
<td>3.35 No clustering of strut lengths</td>
<td>180</td>
</tr>
<tr>
<td>3.36 Algorithm for locating surfaces between bubbles</td>
<td>182</td>
</tr>
<tr>
<td>3.37 Bubble cluster generated with the soap bubble tool</td>
<td>183</td>
</tr>
<tr>
<td>3.38 Pattern matching between sets of points on spheres</td>
<td>184</td>
</tr>
<tr>
<td>3.39 Early sketches for competition design by Chris Bosse</td>
<td>189</td>
</tr>
<tr>
<td>3.40 “Wet model” of bubble pavilions</td>
<td>190</td>
</tr>
<tr>
<td>3.41 Samples of postgraduate workshop outcomes</td>
<td>197</td>
</tr>
<tr>
<td>3.42 Comparison of questionnaire feedback and logged data</td>
<td>199</td>
</tr>
<tr>
<td>3.43 Examples of tool appropriation</td>
<td>201</td>
</tr>
<tr>
<td>3.44 Presentation slide suggesting tool use as medium</td>
<td>201</td>
</tr>
</tbody>
</table>
4.1 Design conversations ........................................... 212
4.2 Mapping discontinuity onto continuity ......................... 219
4.3 Drawing analogies and drawing distinctions ................... 222
4.4 Pre-rationalisation and post-rationalisation ..................... 224
4.5 Descriptive model of novelty-generating processes in design and science ........................................... 236

Volume II – Appendices and Glossary 264

C.1 Execution of Zellkalkül example code ......................... 274
D.1 Eye-disk development in Drosophila melanogaster ............... 277
D.2 Drosophila eye-disk development in Zellkalkül ................. 278
D.3 Cell lineage of Caenorhabditis elegans ......................... 279
D.4 Modelling of Caenorhabditis elegans embryo development ........ 279
E.1 Vertices in rhombic dodecahedron .............................. 282
E.2 Edges in rhombic dodecahedron .............................. 283
E.3 Faces in rhombic dodecahedron .............................. 284
E.4 Vertices in truncated octahedron ............................. 285
E.5 Edges in truncated octahedron ............................. 286
E.6 Faces in truncated octahedron ............................. 287
E.7 Vertices in Weaire-Phelan-like pentagonal dodecahedron .......... 288
E.8 Edges in Weaire-Phelan-like pentagonal dodecahedron .......... 289
E.9 Faces in Weaire-Phelan-like pentagonal dodecahedron .......... 290
E.10 Vertices in Weaire-Phelan-like tetrakaidecahedron ............ 293
E.11 Edges in Weaire-Phelan-like tetrakaidecahedron ............ 294
E.12 Faces in Weaire-Phelan-like tetrakaidecahedron ............ 295
F.1 First panel entered into competition stage I .................... 297
F.2 Second panel entered into competition stage I .................. 298
F.3 First panel entered into competition stage I by Jimenez et al. .... 299
F.4 Second panel entered into competition stage I by Jimenez et al. .... 300
F.5 First panel entered into competition stage II by Jimenez et al. .... 301
F.6 Second panel entered into competition stage II by Jimenez et al. .... 302
F.7 Third panel entered into competition stage II by Jimenez et al. .... 303
F.8 Forth panel entered into competition stage II by Jimenez et al. .... 304
F.9 Fifth panel entered into competition stage II by Jimenez et al. .... 305
G.1 Basic soap bubble tool setup ................................ 307
G.2 Sphere clustering in two dimensions ........................ 307
G.3 Sphere clustering in three dimensions ....................... 308
G.4 Assignment of “virtual bubble pressure” parameters ............. 309
G.5 Generation of irregular space grid .......................... 310
G.6 Sphere packing and derived irregular space grid ............... 311
G.7 Removal of spheres and counting of different strut lengths .... 311
G.8 Import into 3ds Max® ........................................ 312
G.9 Import into ANSYS® and structural simulation ................ 312
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.1</td>
<td>User interface of the soap bubble tool</td>
<td>313</td>
</tr>
<tr>
<td>H.2</td>
<td>“Quick-start card” for soap bubble tool</td>
<td>314</td>
</tr>
<tr>
<td>H.3</td>
<td>“Quick-start card” for tofu cube tool</td>
<td>315</td>
</tr>
<tr>
<td>H.4</td>
<td>Pages 1 and 2 of workshop questionnaire sheet</td>
<td>316</td>
</tr>
<tr>
<td>H.5</td>
<td>Pages 3 and 4 of workshop questionnaire sheet</td>
<td>317</td>
</tr>
<tr>
<td>I.1</td>
<td>Outcome of questionnaire questions 9 and 24</td>
<td>326</td>
</tr>
<tr>
<td>I.2</td>
<td>Outcome of logged functions</td>
<td>326</td>
</tr>
<tr>
<td>L.1</td>
<td>Pills representing complex particle systems</td>
<td>334</td>
</tr>
<tr>
<td>M.1</td>
<td>Three categories resulting from the theory of notationality</td>
<td>344</td>
</tr>
</tbody>
</table>
## Volume I

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Strut length minima in dodecahedral structure</td>
<td>152</td>
</tr>
</tbody>
</table>

## Volume II – Appendices and Glossary

| E.1 | Vertex coordinates in rhombic dodecahedron | 282 |
| E.2 | Edge connections in rhombic dodecahedron | 283 |
| E.3 | Faces in rhombic dodecahedron | 284 |
| E.4 | Vertex coordinates in truncated octahedron | 285 |
| E.5 | Edge connections in truncated octahedron | 286 |
| E.6 | Faces in truncated octahedron | 287 |
| E.7 | Vertex coordinates in Weaire-Phelan-like pentagonal dodecahedron | 288 |
| E.8 | Edge connections in Weaire-Phelan-like pentagonal dodecahedron | 289 |
| E.9 | Faces in Weaire-Phelan-like pentagonal dodecahedron | 290 |
| E.10 | Vertex coordinates in Weaire-Phelan-like tetrakaidecahedron | 293 |
| E.11 | Edge connections in Weaire-Phelan-like tetrakaidecahedron | 294 |
| E.12 | Faces in Weaire-Phelan-like tetrakaidecahedron | 295 |
| I.1 | Sample page of tool logging data | 325 |
Outcomes of innovative designing are frequently described as enabling us in achieving more desirable futures. How can we design and innovate so as to enable future processes of design and innovation? To investigate this question, this thesis probes the conditions, possibilities and limitations of toolmaking for novelty and knowledge generation, or in other words, it examines designing for designing.

The focus of this thesis is on the development of digital design tools that support the reconciliation of conflicting criteria centred on architectural geometry. Of particular interest are the roles of methodological approaches and of biological analogies as guides in toolmaking for design, as well as the possibility of generalising design tools beyond the contexts from which they originate. The presented investigation consists of an applied toolmaking study and a subsequent reflective analysis using second-order cybernetics as a theoretical framework.

Observations made during the toolmaking study suggest that biological analogies can, in informal ways, inspire designing, including the designing of design tools. Design tools seem to enable the generation of novelty and knowledge beyond the contexts in and for which they are developed only if their users apply them in ways unanticipated by the toolmaker. The reflective analysis offers theoretical explanations for these observations based on aspects of second-order cybernetics. These aspects include the modelling of designing as a conversation, different
relationships between observers (such as designers) and systems (such as designers engaged in their projects), the distinction between coded and uncoded knowledge, as well as processes underlying the production and the restriction of meaning.

Initially aimed at the development of generally applicable, prescriptive digital tools for designing, the presented work results in a personal descriptive model of novelty and knowledge generation in science and design. This shift indicates a perspective change from a positivist to a relativist outlook on designing, which was accomplished over the course of the study.

Investigating theory and practice of designing and of science, this study establishes an epistemological model of designing that accommodates and extends a number of theoretical concepts others have previously proposed. According to this model, both design and science generate and encode new knowledge through conversational processes, in which open-minded perception appears to be of greater innovative power than efforts to exercise control. The presented work substantiates and exemplifies radical constructivist theory of knowledge and novelty production, establishes correspondences between systems theory and design research theory and implies that mainstream scientific theories and practices are insufficient to account for and to guide innovation.
Outcomes of design and innovation are described as enabling us in achieving more desirable futures. How can we use design and innovation to enable future processes of design and innovation? To investigate this question, this thesis probes the conditions, possibilities and limitations of toolmaking for novelty and knowledge generation, or, in other words, of designing for designing. This chapter establishes the self-referential nature of toolmaking and its role in human history. It also outlines the scope, the structure and key terminology of this thesis.
0. BACKGROUND

0.1 Introduction

The capacities of language use, that of designing (see Heskett [2002], pp. 8-9) and those of making and of using tools (see Baber [2003], pp. 3-4) are frequently cited when humans discuss the distinguishing characteristics that set us apart from other animals. Tattersall [1998] describes human evolution as proceeding through the crucial steps of toolmaking, the adoption of language, and with it, symbolic thought, and subsequently the occurrence of inventiveness and originality, the combination of which he suggests defines us as humans. The identification of humans with their toolmaking capacity as described by the term “homo faber” has resulted in the classification of Homo habilis as the first human species, largely because it was thought to represent the maker of early rudimentary stone tools (ibid., p. 126). According to Tattersall (ibid., p. 134), there is no question that the invention of (stone) tools represents “a major hominid life-style change, as well as a cognitive innovation of enormous consequence”.

An effort to support the design of design tools, as the one presented in this thesis, must enquire the nature of novelty-generation as well as the role tools play in it. But these are not necessarily easy-to-understand subjects. On the one hand, tool making can be observed as a continuous, self-referential process, in which new tools are accomplished through the application of existing ones. On the other hand, Tattersall (ibid.) describes the innovation of the earliest tools as discontinuous and as resulting in a major cultural transition. What is the nature of this process? It can be assumed that on at least one occasion, the use of a stone as a tool must have come about through some process other than imitation. But was the very first stone tool a found object that happened to display useful characteristics, or was it a made object that was shaped intentionally to display useful properties? In other words: was the first stone tool discovered or was it invented?
Following millions of years of use of largely unaltered types of stone tools that were achieved through imitation, it was, according to Tattersall (ibid., p. 28), not until the appearance of language and of symbolic thought in *Homo sapiens* that the production and application of tools became subjects to the inventive imagination and creativity we take as characteristic of our species. Eventually, tools and the processes from which they originate became the focus of experimentation. Observations made during the experimentation, in turn, became the subjects of design-philosophical reflective analysis. In this thesis, I engage with the full depth of this abstraction hierarchy. But despite all abstraction and despite the remoteness of philosophical reflection from the circumstances in which the first tools were made, my aim is to reflect this subject as what it essentially is: as human. As toolmaking, language and curious inventiveness are described as central to human nature, this report of a study into the design of design tools addresses the core of what we believe defines us as humans.

The example of language use shows that it is one thing to acquire and to make use of a capacity but that it is quite another thing to be able to reflect upon such a capacity and to affect the way it might support others who use it. One may be a proficient speaker of the English language and yet find it challenging to write a book on English grammar. Similarly, while everyday tool use and designing may be experienced as commonplace and natural, enquiry into toolmaking for design may not be experienced as equally straight-forward. The reason lies in the self-referential nature of both activities. Writing on grammar and the design of tools deploy as their means the very object of their enquiries. Just as everyday language use is a prerequisite for writing a book on grammar, the ability to design precedes the ability to reflect upon and to invent tools for designing. At the level of doing research something very similar applies as we are ourselves involved
in our processes of observation (see Foerster [2003], p. 285). Imagine studying a microscope with the only observational method available involving the same microscope itself. Similarly, methods for doing research and of designing as well as digital toolmaking are all drivers as well as products of our doing. The concepts and arguments presented in this thesis are therefore centered on self-reference.

During the past half century, the applied designing of digital tools to support designing has come a long way in becoming a well-established element of advanced design practice in the architectural field. For a number of years digital toolmaking for design has been, and continues to be, discussed in numerous published case studies. The vast majority of case studies seem to have in common that their efforts to support digital toolmaking are undertaken within the contexts of the respective design projects and by the projects’ designers themselves or in conjunction with associated experts. Leading architectural offices such as Gehry Partners and Foster and Partners have established in-house digital tool development support and integrated their operations into the offices’ design development processes, while engineering firms such as Arup are joining design projects already at the earliest stages. From this “inside perspective” a new kind of digital crafts-person has appeared who supplies design projects with specific tools as needs develop in the context of applied design projects.

This pattern differs from the approach taken to produce software tools for purposes other than designing. Text processors, digital drafting tools, email clients and image processing software are developed well in advance of and outside of the immediate contexts of their application. These tools are also generic, in that

---

1 Recent editions of such case studies include Kolarevic [2003], Leach et al. [2004] and Chasznar [2006].

2 Some of these approaches are discussed in greater detail in section 1.4.3
they are applicable to larger groups of users who can apply them repeatedly to the types of problem they were designed to solve. This is similar to the field of mechanical toolmaking. Hammers, screwdrivers, band saws, chopsticks and cranes are applicable to classes of problems rather than just to single, unique problems. Also, they are not only designed and produced within the contexts of their application. Their generic nature moreover allows the efficient mass-production of large numbers of identical tools to serve many users. From an economical standpoint, this conventional model of toolmaking is comparatively resource-efficient and hence seems to be preferable to the design and production of tools just-in-time and in the context of their application. Mechanical and digital tools are oftentimes observed to enhance productivity with regard to some intention in a given context. In design contexts, which are notorious for tight deadlines and limited resources, the requirement for productivity is well-known to practitioners. Nevertheless, the production of design tools seems to resist the efficiency offered by the conventional toolmaking approach in other areas. Aiming to explore the differences between design and other areas and whether digital design tools can be used by wider groups of users, I take the toolmakers’ perspective to investigate the practice of making digital design tools, its conditions, constraints and its possibilities, while keeping the context of the tools’ application firmly in sight. I hope this thesis is of interest to those who support designing, including digital design toolmakers.

---

3 To this date, there is no standard tool for developing new ideas in say architecture in the way in which other tasks can rely on standard tools such as screwdrivers for tightening screws, knifes for cutting things and so forth.
0.2 A personal perspective

To contextualise the work presented in this thesis, I shall note two points regarding my own background. Firstly, I received no formal training as a designer prior to this study. Secondly, the background and education I received before this project at an electrical engineering high school and academic training as a teacher were largely centred on formal procedures, and as I realised through this study, were based on a positivist world view. This world view assumes the existence of an objective reality and prioritises methodical processes, formal specifications and rational reasoning over subjective viewpoints, differences in individual intuition, and the values of ambiguity and uncertainty. There are brief reports on two essential insights I have gained over the course of this study in appendices L and M, through which I began to distance myself from the positivist’s outlook during the course of this study.

0.3 Scope and limitations of this thesis

This thesis takes a research-through-design approach to investigate the conditions under which digital design tools are made and used. It also examines the possibilities of informing those who make digital tools for designing in their efforts. The thesis initially ventures to develop a prescriptive, generally applicable digital tool for designing. It however arrives at a personal model of novelty and knowledge generation, which accommodates theoretical concepts regarding novelty and knowledge generation in science and design others have put forward.

The scope of this study as well as its applicability and validity are limited primarily due to a contradiction between theory-building research, which aims at the production of generalisable knowledge and applied designing, which is concerned with concrete design challenges. Another limitation results from
the presence of ideologically inspired value judgements in designing and the avoidance thereof aspired for in academic research. Designing and intentional interferences with our internal and external environments are driven by curiosity and interest in some form of progress, or the achievement of somehow more desirable futures. In this sense, the study presented in this thesis assumes that novel things and new knowledge are worthwhile striving for. Motivations of this kind involve subjective value judgements, which are difficult to subject to formal reasoning. These are taken as given in this thesis and are not challenged or questioned.

Processes and outcomes of designing are not deterministic. They oftentimes involve surprise and contradiction. Statements about designing are thus likely to be challenged, limited or negated by some observation or argument. This also applies to statements about designing put forward in this thesis, which assume that initiatives to support and to enable designing lead to better designing while control and restriction of designing lead to worse designing. These assumptions, again, cannot be argued about formally – some authors even describe scenarios in which the opposite seems to apply. The theoretical body used as a framework in this thesis named second-order cybernetics and the overall argument put forward in this thesis favour freedom and absence of control over restriction in designing. While it is often assumed that tools need to offer utility that can be determined and assessed by toolmakers (in analogy for example to professionals such as physicians

Some design educators argue that the educational application of digital design tools with restrictive properties can be desirable, or, conversely, that excessive freedom can be confusing for novices. Achten et al. [2000], pp. 455-456 for example argue that simple tool functionality allows fast learning suitable for introducing first-year architecture students to a basic understanding of computer-aided architectural design.
or architects who “know what is best” for their clients; this is the position I initially took in this study), this thesis limits the assessment of utility of design tools to the post-hoc opinion of design tool users.

The work and arguments presented in this thesis are centered around an applied toolmaking study, which takes a research-through-design approach. Within the toolmaking study, research is limited to the design of two- and three-dimensional space grids and the reconciliation of contradicting geometrical requirements. The toolmaking study was not a rigorous formal process but one which aimed for “good enough” outcomes based on subjective reasoning in given situations. The toolmaking study is then reflected upon from the perspective of second-order cybernetics, which, from my point of view, offers good explanations for observations I made during the toolmaking study. From these explanations and from some theoretical concepts proposed by others I then develop a model of novelty and knowledge generation in science and in design. The analogies and distinctions drawn in this process of inference are subjective. The goal of this thesis is to make the process of inference traceable and comprehensible for others, who may draw their own conclusions from this thesis, not to argue for general validity and applicability. The proposed model is descriptive as opposed to prescriptive and the presented arguments are personal. Others would have observed, inferred and argued differently.

The discussed limitations of this study result from theoretical positions I have learned and adopted in and through this study. They are therefore at the same time outcomes of this study and discussed in depth in chapter 4.
0.4 Structure of this thesis

This thesis consists of two volumes. Volume I contains “front matter”, the main body of the thesis and the references list. Volume II contains supporting material in appendices and a glossary. The main body consists of the following five chapters:

Chapter 0 gives an introduction to the work presented in this thesis and outlines its context. It presents its scope and describes the basic structure of the chapters in the thesis document. It further outlines key terminology used throughout this thesis.

Chapter 1 reviews literature and examines previous work by others in fields related to this study. It lays the foundation for the overall structure of the argument presented in this thesis, that is, the account of a hands-on toolmaking study and a subsequent reflection from a second-order cybernetic perspective. The first three sections of chapter 1 discuss previous work and findings in the fields of tools and computer-aided design, generative design and space grid geometry. The purpose of these sections is to establish and familiarise readers with the context, common practices, terminology and key challenges which are relevant to the toolmaking study presented in chapter 3. The following section of the review of work by others discusses key positions and challenges of epistemology in the context of science and of design. This discussion lays out the theoretical framework for the reflective analysis of the toolmaking study presented in chapter 4. Chapter 1 concludes with a brief summary and with the primary research questions investigated in subsequent chapters.

Chapter 2 outlines the strategic approaches adopted to investigate the research questions posed at the end of chapter 1. It outlines research strategies at two levels: The first level is concerned with the design of space grids, the design of digital tools for space grid design as well the application and evaluation of these tools. It
0. BACKGROUND

outlines the study summarised in chapter 3. The second level is concerned with the possibilities and the limitations of the processes reported in chapter 3. It describes the second-order cybernetic framework applied to reflect observations made in chapter 3 in chapter 4.

Chapters 3 and 4 present the main body of research work undertaken in this study. This research work takes the shape of applied designing and a subsequent reflection on the design process. Driven and structured by a design process, the outlined work is not directed at one set goal. It aims at a number of different goals at different times, according to insights gained during the process of the enquiry. This process starts out aiming for the development of a prescriptive digital tool for designing and it concludes presenting a descriptive model of innovation in design and in science.

Following a short introduction, chapter 3 discusses ten successive digital design toolmaking exercises I carried out in the context of this study, and which are concerned with the design of space grid structures, the design of digital tools for space grid design as well as the application and evaluation of these tools. Chapter 3 concludes with a brief summary.

Chapter 4 presents a reflective analysis of the toolmaking study described in chapter 3. This analysis uses epistemological positions reviewed in the literature review in chapter 1 as a theoretical framework, with an emphasis on second-order cybernetics. I draw upon seven aspects of cybernetic theories of knowledge production to present key findings that offer explanations for observations made in the preceding toolmaking study. Chapter 4 further summarises and concludes the discussion of observations and lays out contributions of and limitations to this study, and ends with a section on future work.

The appendices include recorded expert input, supporting documentation on
tools developed in this study, exercises and evaluations carried out (including
the participation in a design competition and a postgraduate design workshop)
and detailed geometric analyses of polyhedra underlying some of the space grids
investigated in this study. While appendices L and M do not fit into the structure of
the main volume of this thesis, I find it necessary to include them to explain shifts
in my perspective over the course of the study.

0.5 Terms used in this thesis

This section is intended to briefly explain, discuss and relate key terms used
throughout this thesis. Additionally, the glossary in appendix N lists and explains
geometry-related terms. The purpose of this section is to prepare readers for the
flow of the argument in later chapters while the purpose of the glossary is that of a
quick reference list to make technical descriptions and arguments comprehensible
to those who are not familiar with the terminology of computational geometry,
which is drawn upon in particular in the toolmaking in chapter 3.

In this thesis, design is understood as described by Glanville [1994b], [1996] as
the making of the new, a conversational, novelty-generating, hence knowledge-
generating process in response to challenges that are inherently difficult to de-
fine. Of particular interest in this context is the attempt to reconcile conflicting
requirements. Novelty generation is understood synonymously to innovation.
Knowledge is understood as the sum of beliefs, which are assumed to be justified
by its owner(s) (see Downton [2003], p. 57). Knowledge can be presented in
various (including communicable) ways. The understanding of design as a process
of knowledge generation refers to the observation that knowledge grows during
design processes and new knowledge is to some extent presented (externalised) in
the resulting design outcomes.
Tools are understood as man-made artefacts that are external to the human body and applied by somebody to tasks so as to attain somebody’s goals. The function of tools is to enable their users in performing tasks or to offer their users benefits of some kinds when performing tasks, such as increased productivity, precision or safety. Tools are made to perform in specific ways. These specific ways are referred to as methods and tools are thus viewed in this thesis as “solidified” elements of methods. Based on the categorisation of human activity into physical and cognitive activities, tools can be categorised as aimed towards supporting physical or cognitive tasks. Tools can be implemented as software on computers by means of computer programming. One specific kind of programming is (macro) scripting within the context of software applications. In this case, objects and functions of a software application, for example a three-dimensional modelling software, are accessed and controlled from code scripts that are usually programmed in interpreted scripting languages such as Python, PERL, TCL, VBA or ECMAScript (see McComb [1998]) dialects such as JavaScript™. Tools developed and examined in the context of this thesis are primarily a cellular automata modelling application called Zellkalkül, which allows user scripting in an ECMAScript dialect and various VBA macro scripts for MS Excel®, which access and control three-dimensional modelling functions in Rhino3D®. All tools examined in this thesis are software programs aimed at supporting cognitive activities in contexts of designing as it is discussed above, and thus referred to as digital design tools. Tools are the product of novelty generating human thought and action and thus represent encoded knowledge. The activity studied in this thesis, the making of tools for designing, is hence understood as designing for designing.

The term parametric design is used in this thesis to describe a computer-aided geometric modelling and design exploration approach. In this approach,
which is also referred to as **associative geometry**, geometric entities such as points, lines, surfaces and bodies as well as relationships between them are, either partially or entirely, defined in an abstract fashion, representing generic **geometric topologies**. These topologies can produce instances of geometric form when sets of parameters are applied to specify necessary counts and measures. The principle can conceptually be related to object-oriented programming, in which abstract classes are defined that can be instantiated to derive specific objects. This approach can for example be found in three-dimensional modelling environments, in which geometric primitives such as spheres and boxes are typically parametric models that are instantiated with additional specific dimensions, position and rotation data. In parametric design, models are usually more complex, and topologies can be defined by users. Parameters can further be associated to create dependencies or conditions. A simple example is a set of four straight lines, connected at right angles at four points to form the topology of a square. If this parametric model is set up to receive a numerical parameter to specify the length of all four lines, then, applying the parameter $5cm$ will result in a square with an edge length of $5cm$. If a similar topology were set up to receive multiple parameters for individual edge lengths and angles, not only squares but also many other four-sided polygons could be instantiated, and possibly considered and evaluated within a design context. This possibility of conveniently producing variety in design geometry makes parametric design an interesting technique for design exploration. Obviously, not all combinations of parameters and not all topological relationships are suitable to produce geometrical form. A four-sided polygon with the four edge lengths $1cm$, $1cm$, $1cm$, $20cm$, a triangle with a sum of its angles other than $180^\circ$ or a polygonal topology defined by $n$ edges joined at $\frac{n}{2}$ points are examples of shapes that, geometrically, cannot exist. Topologies that are set up
in ways that prohibit the production of geometric form are referred to as **over-constrained**. To reduce potential geometric forms to desired or valid variations or variation ranges, **parametric constraints** can be defined with the set-up of the geometric topology. A brief review of parametric tools used in this design approach is given in section 1.3.2.

The term **evolutionary design** is used in this thesis to describe computational design approaches that use some form of **genetic algorithm**, aiming to mimic form-producing processes of biological evolution. In this approach, elements of form such as parts, geometric elements, topological relationships and geometric parameters are mapped onto, and are determined by, **genetic codes**. These are sets of digital data which are used to mimic the function attributed to DNA in evolutionary theory. This entails a duality of a digital code and a formal manifestation of the code in analogy to genotypes and phenotypes described by evolutionary biology. Variation of the digital genetic code (genotype) results in variation of the form (phenotype) and is achieved by operations modelled after those observed in biological evolution, namely inheritance, mutation, selection and crossover. The application of these operations can be automated to produce **populations** of design options. Outside of the design field, evolutionary algorithms are frequently applied as a search technique, whereby so-called **fitness functions** are specified, allowing not only for automated production of variety but also for automated selection from variant populations, usually leading to **convergent** processes of optimisation. In design, where **divergent** explorations are of interest, human selection is oftentimes employed instead of automated selection. A brief evolutionary design tool review of this design approach is given in section 1.3.3.

Evolutionary design describes computational design approaches that utilise genetic algorithms to mimic form-producing processes of biological evolution.
In analogy to biological evolutionary processes, evolutionary design introduces variety into “genetic” symbol sequences (or “digital DNA”) by means of genetic operators, namely mutation, crossover and selection. During evolutionary design processes successive generations and populations of design variants are then derived from the manipulated symbol sequences based on (predefined) “genotype-phenotype mapping”. The resulting design variants are evaluated as candidate solutions and either selected, discarded or subjected to further genetic modification in analogy to evolutionary theory in biology. In each generation, performance criteria encoded in fitness functions can be automatically evaluated to choose more appropriate variants for further “breeding” of subsequent generations of candidate design variants.

The term cellular development is used in this thesis in reference to the field of biological development. This field investigates biological processes of form expression in higher organisms. At the start of the cellular developmental process stands a single cell called the zygote. By repeated splitting, it produces a family tree called a lineage and thus generates multi-cellular tissues, which, in a process of differentiation develop into different types of tissues so as to express for example different organs. The developmental process is determined by the genetic code (DNA) in interaction with environmental influences. In this thesis, the principles of developmental biology are not examined at a formal or in-depth level but referred to as an inspiration for computer-supported form finding processes, in which “virtual cells” develop by means of “digital DNA scripts”.

Cellular automata were introduced by mathematicians who used them to investigate self-replicating systems, during the late 1940s (see Schrandt and Ulam [1970]). Cellular automata are arrays of cells, which are usually arranged on regular Cartesian lattice grids in one, two or three dimensions. Cells can assume different
predefined states according to rules, which are typically few and simple, and which express how each cell’s state should change according to the states of cells in the immediate neighborhood. Cells perform their rules at discrete time intervals, generating dynamic and to some extent non-deterministic overall patterns.

In the study of two-dimensional geometry, a polygon is a plane figure bounded by a closed path of straight edges, which are joined at vertex points. Similarly, in the study of three-dimensional geometry, a polyhedron is a body with a surface composed of flat polyhedra, joined by their edges. The surfaces are called the faces of the polyhedron and the points where edges meet are called vertices. Packed polyhedra can fill space. Their topology of vertices, edges and faces is also referred to as a lattice or space grid. Edges within lattices are sometimes also referred to as vectors. In a convex polyhedron, a line connecting any two points on any two different faces lies entirely inside the polyhedron. A triangulated approximation of the surface of an orange for example is a convex polyhedron while a triangulated approximation of the surface of a banana is not. As a banana is partially concave, the line connecting both ends of the banana shape would lie outside of its body.

In three dimensions, the vast majority of space grids is based on periodic repetition in Cartesian order\(^5\). That is, they replicate periodically along mutually perpendicular x, y and z axes, which also results in a high degree of symmetry. Periodicity and symmetry, if present in grid structures, result in finite (possibly small) numbers of required elements such as edge lengths and vertex angles, facilitating rationality in respective space grid manifestations, thus allowing

\(^5\)A known exception is the so-called Danzer tiling (see Danzer [1989]), which has been described as a three-dimensional analogue of the two-dimensional Penrose tiling (see Ramachandra et al. [2000]).
efficient batch production of components and assembly procedures.

This thesis distinguishes the terms space grid and space frame. Grids are tessellations of two- or three-dimensional space that are often used as ordering frameworks and proportioning systems in design, where they allow the establishment of relationships between parts and wholes. Areas of their design application range from typographic design to urban planning. For these purposes, the use of grids usually aims to break down space in regularly ordered ways, involving limited sets of proportions, angles and distances. Space grids are geometric topologies defined by sets of curves (usually straight lines), which form two- or three-dimensional network structures by means of sharing start and endpoints. Connected curves in a space grid, which form loops, can describe surfaces such as flat polyhedra and multiple such surfaces can describe polyhedral cells, which resemble clusters of soap bubbles or cells in tissues. Thus, space grids can describe cellular tessellations of space. Space frames are built architectural structures consisting of networks of struts, which are usually made of steel or wood, that act as tension or compression members; and joints between them, which are oftentimes executed as distinct node elements. Three-dimensional space grids can be translated into architectural space frames. In this case, the curves of a space grid are interpreted as space frame struts, and the points at which curves meet in a space frame are interpreted as locations of space frame nodes. The abstract nature of space grids does not suffice to fully describe space frames, as space grids do not address issues such as material choices or joint detailing. Space frames allow the even distribution of stresses within large, light-weight structures. In related structural engineering contexts they are sometimes also referred to as “space trusses”. Space frames can be described as circumscribed packed polyhedra and they can in many cases be derived from packings of spheres as shown on the
right of figure 1.13. There are two basic ways in which packed spheres can be translated into polyhedra and space frames. The first one simply connects centre points of adjacent spheres by what Fuller [1975], p. 155 calls “radial vectors” as shown on the left of figure 0.1. The second one first expands the spheres to form space-filling polyhedra and then replaces the polyhedral edges with what Fuller (ibid.) calls “circumferential vectors” as shown on the right of figure 0.1.

Figure 0.1: Translation of a packed sphere into radial (left) and circumferential (right) vectors.

This relationship between spherical and polyhedral packings and space frames becomes visible in the shapes of soap bubbles. The study of the geometry of soap bubbles therefore offers insights into possible strategies for the design of tessellated structures, in particular as they exemplify processes and results of surface minimisation and thus lightness, as well as of irregularity based on uniform physical laws.

Figure 0.2: Lattice grids of rhombic dodecahedra (left), Kelvin-like (centre) and Weaire-Phelan-like (right) polyhedra. 18 cells each, rendered in parallel projection at identical scales.

Gabriel [1997] has edited a volume on the relationship of polyhedra and space frames with an architectural focus.
Findings in the study of polyhedral space tessellation and close packing have applications in diverse areas including material analysis in metallurgy, cellular morphology in biology and architectural design involving minimal surfaces. The three-dimensional polyhedron with the smallest number of faces and edges is the four-faced tetrahedron while there is no upper limit to the number of faces, edges and vertices of a polyhedron. Polyhedra with high counts of faces, edges and vertices can take highly elaborate and detailed shapes. The vector-oriented three-dimensional models in computer graphics, for example, or stereolithography (STL) data, consist of triangular faces joined at their edges to form polyhedra, which can approximate virtually any kind of volumetric shape. Theoretical analysis of polyhedra, however, being concerned with their underlying order, explanatory principles and laws, focuses largely on the smaller and the more simple polyhedra with face-counts of below 20 to 30, such as four-faced tetrahedra, six-faced cubes or twelve-faced pentagonal or rhombic dodecahedra. A class of polyhedra considered most basic is called regular polyhedra. Each of them consists of congruent faces only (the cube, being one of them, consists of six identical square faces). Regular polyhedra are also defined as containing only one edge length and only one vertex angle (the angle at which edges meet at a vertex). There are only five such polyhedra: cube, pentagonal dodecahedron, icosahedron, octahedron, and tetrahedron. They were later described by Plato and are hence known as the set of **Platonic solids**. A second class of polyhedra, which was known to the ancient Greeks and described by Archimedes, again with only one edge length, shows a

---

7Non-triangular polygons can be triangulated by adding one or more vertices inside the polygon. The surface of any polyhedron (consisting of polygons joined by their edges) can be triangulated by adding one or more vertices in the plane of each non-triangular face.
similar arrangement of two or more different faces arranged in the same way about each vertex. This class is known as the set of Archimedian solids. An overview of Platonic and Archimedian solids is given in Nooshin et al. in Gabriel [1997], 355. All of these polyhedra are convex polyhedra (this term is explained in section 0.5).

Without the properties of symmetry and periodicity, packed polyhedra can still fill space in less regular fashions. An increasing number of architectural designs aims at this kind of visual irregularity, which consequently involves penalties with respect to efficient manufacturing, analysis and so forth. In this context, visual irregularity denotes the level of difficulty at which an observer is able to identify repetition in a given structure.

For building brick houses, for example, the basic brick can be prefabricated identically in large numbers and used in as many brick houses as required. This is economically advantageous since neither do the bricks within one building require variation of any kind nor is there any variation of bricks necessary between individual houses. The bricks can be mass-produced from identical molds off-site and are interchangeable. In this sense, brick masonry can be considered “rational”.

The design operation for which this thesis investigates digital design tools is geometry rationalisation in space grids. This is a special case of design rationalisation. Space frames, and buildings in general, are rarely made up of only one single system of elements or of one single integral physical object (as noted by Aish [2005], p. 11). It is similarly rare for a building shape to consist of one or very few primitive geometric shapes and for this reduced geometry to be at the same time consistent with all other building aspects such as structure or the programming of uses. Buildings are usually comprised of multiple sets of building elements and integrate various geometric shapes. These elements must serve various needs and intentions. These include the achievement of an overall
expressive gesture, the requirements for building services and those imposed by structural necessities, underlying fabrication processes as well as the procedural constraints imposed by construction logistics. These diverse intended functions must be organised within themselves and coordinated amongst each other despite their oftentimes conflicting relationships. The necessary process of conciliating conflicting aspects of a design project, as well as the outcome of this process, are referred to in this thesis as design rationalisation (see also Fischer [2007]). In the case of geometrically complex architectural design the process of identifying and developing geometric solutions that mediate between conflicting intentions is referred to as geometry rationalisation (see Fischer ibid.). This process involves the identification of strategies for breaking down surfaces and volumes into tessellations or other rational descriptions of some kind of geometric order as well as the development of manufacturing procedures for producing suitable related building components and methods of assembly.

As space frames consist of struts and nodes at which the struts meet, the rationalisation of visually irregular space grid geometry aiming for cost-efficient (repetitive) manufacturing and construction procedures seeks to reduce the number of different strut lengths and the number of different node angles.

In this thesis I use the word self-reference to describe the condition of an entity (such as a human, a machine or comparable system) having an effect on itself. The reliance of toolmaking on previously designed tools alluded to in the title of this thesis is an example. Another word for this condition is recursion. In this thesis, I assume that designers use their knowledge and resources they find in their environments to create novelty, which is essentially new knowledge. Through the knowledge-generating process of designing, designers transform themselves. Designing is thus a self-referential process, which cannot sufficiently be described
in terms of linear causality. **Cybernetics** is a field of study concerned with communication, feedback and control in systems of all kinds including machines and living organisms. The term is also used in reference to the body of theory developed in this field. **Second-order cybernetics** is, as its name suggests, based on cybernetics, of which it is considered to be the more general and more powerful case (see Glanville [1997], p. 6, footnote 7). It differs from cybernetics by taking into account the observer who, in observing observables (including systems displaying feedback and control) becomes part of another system (see ibid., p. 7). **Second-order cybernetics** recognises that observers can observe themselves and that they can observe and recognise things that have effects on themselves. It is thus capable of describing self-referential conditions such as those of designing.

The terms **analogue** and **digital** refer to the distinction between (analogue) continuous forms of representation, such as for example seismograph readouts, and those forms of (digital) representation that are based on non-continuous codes, such as for example telephone numbers. In this thesis, the terms analogue and digital are also used to distinguish between computer-supported (digital) processes and non-computer-supported (analogue) processes. The reflection of this thesis in chapter 4 uses the two words in terms of ways of presenting knowledge. Appendix M gives a more detailed discussion of the terms analogue and digital and cites an appendix from Wilden [1980], pp. 191-195, which presents an extensive examination of this distinction and its observer-dependence.

This thesis also refers to the **drawing of distinctions** and to the **drawing of analogies** as fundamental cognitive activities. The drawing of distinctions is used as described by Spencer-Brown [1997] and modified by Glanville and Varela (see Glanville [2002]), which can, in a straight-forward way, be explained as the process

---

8These examples were put forward by Goel [1995], p. 166.
by which somebody arrives at the statement “this is different from that”. The
drawing of analogies is used as described by Lakoff and Johnson [1980], which
can, in a straight-forward way, be explained as the process by which somebody
arrives at the statement “this is like that”.

This chapter reviews previous work by others relating to toolmaking for novelty and knowledge generation (i.e. designing for designing). The review of previous work on toolmaking covers anthropological studies of toolmaking and tool use in humans and other animals, approaches of engineering design as well as the fields of design methods and CAAD research. The review of previous work on novelty and knowledge generation covers the role of Nature as a source of human inspiration, the field of space grid geometry as an application domain for studies presented later in this thesis and models of knowledge growth in science and design. The chapter concludes with the primary research questions investigated in the remaining chapters.
1.1 Introduction

This chapter presents a review of previous work of others to contextualise and to explain key concepts of the work presented later in this thesis. It is, in contrast to the remaining chapters, consciously not developed around a unifying narrative. This chapter reviews literature and work of others in a broad range of different research fields, some of which have influenced my thinking and doing before the design toolmaking study presented in chapter 3 and some of which have only influenced my thinking and doing after it (leading to the reflective analysis presented in chapter 4). This review serves to demonstrate the variety of the fields in which I have conducted research and their philosophical viewpoints. The narrative that connects these fields is the one presented in the two chapters 3 and 4, which outline my own research work. The purpose of sections 1.2 to 1.4 is to establish the context, approaches, terminology and challenges of the toolmaking study presented in chapter 3. Section 1.2 reviews key positions regarding the production and use of tools, section 1.3 outlines biological analogies found in generative design approaches, inspirations underlying my earlier research as well as warnings that have become apparent regarding both the powers and limitations of natural and biological analogies in designing. Section 1.4 then presents previous work of others on space grid design with an emphasis on polyhedral packings and soap bubble geometry, the design of architectural space frame structures, the practice of architectural geometry rationalisation and the distinction between pre- and post-rationalisation. Section 1.5 reviews the relationship between design and science with respect to knowledge generation. It discusses some cases of scientific research work involving geometric innovation, for which no suitable methods of enquiry have been readily available. The section concludes with a brief summary of second-order cybernetic theory, which establishes the theoretical framework for
1. REVIEW OF LITERATURE AND WORK OF OTHERS

the reflective analysis of my toolmaking study presented in chapter 4. The purpose of this review of previous work of others is not to give an exhaustive account of work previously undertaken in related disciplines, but to provide a foundation for reporting the limits I have encountered. It also identifies the insights I have gained over the course of this study. This review draws material from three different kinds of sources:

1. Published literature including research monographs and journal articles is analysed in three general domains: tools and tool design, space grid geometry and the relationship between models of knowledge generation in design and in science.

2. The discussion of tool design and space grid geometry includes design case studies of specific software programs and architectural designs. These case studies draw exclusively on accessible published sources in books, journals and magazines and are referenced accordingly.

3. Personal exchanges with experts in related fields. A few items in this review are not found in print in the public domain. They are yet referred to here as I consider them worthwhile mentioning in this review. These include statements that were articulated during question and answer sessions on conferences, remarks professionals made during my industry visits or as parts of public lectures, and personal email exchanges with experts in related fields. In the case of quoted email exchanges, copies of these emails are included in the appendix of this thesis.

The last part of this chapter is a short summary that concludes with the primary research questions addressed in later parts of this thesis that are derived from this chapter.
1.2 Tool use, tool making and computer-aided design

This section reviews key positions regarding the production and use of tools generally as a human capacity and specifically in the overlapping area between design research and software engineering, in which the computer-aided design field finds itself.

1.2.1 Tool use and tool making in humans and other animals

Arguing that the term "tool" has been overly stretched to include supports for all kinds of physical and intellectual human action, Baber [2003], p. 4 defines tools as physical objects that are “manipulated by users in such a manner as to both affect change in some aspect of the environment and also to represent an extension of the users themselves.” Baber proposes that tools are developed gradually rather than designed (ibid., p. 69). He suggests that this process is of an “evolutionary” nature, with variations of tools’ concepts competing for best adaptation to specific functions or users. He applies this description, however, mainly to tool development processes preceding the Industrial Revolution and mass customisation. When discussing the requirements of tools developed for mass customisation, Baber (ibid., p. 70) describes characteristics for “well-designed” tools that are based primarily on ergonomics. According to Baber (ibid., p. 71) tools should, as far as possible, be designed for special purposes rather than for multifunctional use; with the primary aim for tool design being to produce tools that feel comfortable and balanced when held. To achieve this aim, he recommends design principles such as the minimisation of repetitive finger action and minimisation of device vibration and impact forces. He thus implicitly prioritises efficiency both in immediate tool applications as well as in the economy of tool mass production, which allows tools to be designed once by few and then
used time and again by many. Baber (ibid., p. 103) ascribes “failure” in tool use to two reasons: human error and neurological impairment, both of which prevent users from applying tools as they are intended to be used.

Toolmaking behaviour is common amongst but not unique to humans (see Morowitz [2002], p. 155). As tool use plays only a small role in the life of a small number of species, Alcock [1981], p. 231 notes that the attention received by tool use in animals is primarily due to humans’ interest in their own use of tools. In his attempt to give an exhaustive account of cases of tool-using animals in “Animal tool behaviour”, Beck [1980] defines tool use as “the external employment of an unattached environmental object to alter more efficiently the form, position, or condition of another object, another organism, or the user itself when the user holds or carries the tool during or just prior to use and is responsible for the proper and effective orientation of the tool.” (see Beck [1980], p. 1)

Alcock (ibid.) however points out that it is impossible to define the term “tool use” in a simple way. In Beck’s definition, an animal’s faeces, if thrown at an enemy, can be classified as a tool. While the use of found objects as tools in animals is well-known, McGrew [2004], p. 1046 adds that the manufacture of tools is far less common, observed only in a few bird species and in great apes. Of these, only chimpanzees and orangutans are known to have tool kits – repertoires of different tool types – while chimpanzees are known to use two or more different types of tools in sequence to achieve goals. Among apes, toolmaking and tool use differ between populations,
with some behaviours being universal, some regional, and some specific to a single
group. A population’s tool kit can thus be described as a “technological profile
of its material culture” (ibid., p. 1046). From this perspective, the focus on objects
used as tools changes to a focus on processes of tool use and toolmaking. Lestel and
Grundmann [1999] argue that Beck’s [1980] notion of tools is ambiguous and based
on debatable assumptions, and ultimately inadequate when explaining technical
behaviour in animals (ibid., p. 368). Instead, they suggest replacing a theory
of tools with a theory of mediations of actions in animals, which gives attention
not to the objects used as tools but to techniques as social behaviours. Lestel and
Grundmann define mediations of actions as “those material or intellectual aids that
enable an animal to alter its performances or competences, either by changing the
nature of its operation or by increasing its field of action” (ibid., p. 372).

This new perspective allows tool use to be examined as part of more general
strategies developed in the negotiation of social interactions within animal society.
Drawing on Hansell’s [1987] work, Lestel and Grundmann [1999], p. 372 further
propose that tool use depends on “a particular mental attitude that enables the user
to see an object as something other than what it seems to be”. In this view, tools are
objects appropriated by animals to increase possibilities for action, usually within
a social context. This extended notion of tool as mediated action offers for example
an explanation for the cognitive and behavioural changes observed in acculturated
great apes taught by humans to use the rudiments of a symbolic language. Lestel
and Grundmann (ibid.) argue that the extended abilities of such apes beyond the
scope of their wild counterparts could be explained as a cognitive domestication
of the human being by the ape, such that the human serves as an opportunity to
“access cognitive processes normally beyond its reach” (p. 391). In other words,
the ape is using the human teacher as a “tool”.
1. REVIEW OF LITERATURE AND WORK OF OTHERS

1.2.2 Engineering design

In the various fields of engineering, and in particular in mechanical engineering, design is considered and investigated as part of the profession. Dominick et al. [2001], p. 1 state: “In essence, engineers are designers.” Dominick et al. present as “Tools and tactics of design” what might also be referred to as skills and methods used in design contexts. Tools in the sense of physical objects or software for supporting novelty generation are of no significant concern to Dominick et al. Due to the nature of its strategical recommendations, “Tools and tactics of design” can in my opinion nevertheless be considered indicative of how common design engineering positions itself towards issues of support provision for novelty generation. Engineering design emphasises somewhat different aims and qualities of the design process than designers outside the engineering fields do. Its focus tends to be generally less on novelty generation and more on “a process that is high in quality, reliable, competitive in terms of cost, and available in a timely fashion” (ibid.). Accordingly, engineering design emphasises the need for control over the design process as well as over its outcomes. This may be due to a strong focus on safety, manufacturing requirements and predictable performance, as well as on cost-effective solutions. Ullman [2001], p. 6 emphasises the need for “robust products” that are insensitive to “uncontrollable factors” as these “cause the product to operate in a non-ideal, low-quality or noisy manner”. For this reason, engineering design often involves choosing the “optimum assembly of standard components” in order to arrive at its solutions (see Dominick et al. [2001], p. 76). Engineering design typically follows rational procedures, in which initially defined design problems are followed consequently and are systematically broken down into smaller components, which are then considered separately (ibid., p. 37).

The engineering design process is often described as a linear process in which
different aspects of design problems are solved sequentially. To solve subproblems, engineering design usually relies on the application of known tools and tactics and emphasises standardised search and evaluation strategies. Ullman [2001], for example, recommends engineers to follow a design process consisting of 12 steps in order to avoid risk, which he characterises as “the chance that a poor decision will be made” (ibid., p. 12). In Ullman’s perception, risk stems from “uncertainty and the consequences resulting from it”, and reduces the likeliness of achieving the desired goal. Similar to Ullman’s recommended division of the engineering design process into 12 steps, Dominick et al. [2001], p. 5 present engineering design as a process that consists of four phases: defining the problem, formulating solutions, developing models and prototypes, and presenting and implementing the design. Dominick et al. describe the design process as being of an iterative nature, and argue that this iterative nature primarily arises from the need to define problems more clearly as well as from new knowledge or new technology encountered during the design process that change the problem solving approach (ibid., p. 15). According to Dominick et al., most problems engineers have to solve are open-ended and have several possible solutions, and the central challenge is to “identify an optimal solution by careful and systematic analysis of multiple alternatives”.

1.2.3 Software engineering and computer programming

There are several domains in which computers are programmed by experts. The four areas reviewed in this section are the academic field, the corporate field, early “hacking” culture and the open-source community.

The academic programming and software engineering fields are summarised under the name “computer science”. This name, however, is frequently rejected within these fields. Abelson [1994] likens the field to engineering, arts and magic
more than to science. He also points out that the field is not concerned with computers in the same sense in which “biology is not really about microscopes and petri dishes”. DeGrace and Stahl [1991] prefer the name “software engineering” but state that it was just struggling into existence (ibid., p. 5) and is still much unlike a science, due to its inconsistent understanding of its tools and procedures, which vary considerably between different software development contexts and are thus probably most suitably likened to crafts during the Middle Ages. DeGrace and Stahl argue that software engineering must, beyond solving problems, “look into the very nature of problem solving” (ibid., p. 10). They refer to Rittel and Webber’s [1973] notion of “wicked problems” (see also section 1.2.4) and criticise structured problem solving approaches, which inappropriately “assume that the what can be separated from the how” (ibid., p. 67), and that it cannot be applied to a wicked problem (ibid., p. 68).

The development and evaluation of software (design) tools for the production of software is summarised under the name CASE (computer-aided software engineering). Researchers in this field emphasise common “implementation issues” such as questions of technologies and techniques (developer perspective), tool application (user perspective) and tool evaluation. The scenarios which are to be supported by software tools are differentiated according to the scopes of support such as individual tools, workbenches and tool sets and entire environments. Much effort is thus directed not towards the design of tools but towards establishing suitable relationships amongst existing tools. The general criteria for desirable tool development are efficiency in tool production and efficiency in tool use (see Gray [2000], p. 1 and Thomas and Nejmeh [1992], p. 30). Greenberg [1993], for example, regards the computer user as a “toolsmith” who faces the difficulty of using and managing large collections of loosely related tools (ibid., p. 3). Following a brief
review of tool use in animals, Greenberg states that a distinguishing feature of human tool use are the large number and the diversity of tools which are sometimes employed in the same context such as the range of cooking utensils commonly found in kitchens or the range of file manipulation tools commonly found on computer systems. He points out that where human tool-using activity “is hugely dynamic and not readily specified, the choice, and arrangements of loosely related tools cannot be effectively predicted by another person” (ibid., p. 3-4). Therefore, Greenberg proposes organising principles by which tools can be made accessible, based on the analogy of a carpenter’s workshop (see figure 1.2). These principles are: Making most recently used tools most available, arrangement by function and arrangement by task (ibid., p. 4). Colwell [2003] also uses the workshop metaphor, relating his work as a microprocessor designer to his recreational woodwork. He expresses astonishment and appreciation for irrational tool “abuse”, citing the growingly popular use of belt sanders as competitive racing vehicles (ibid., p. 9).

In corporate contexts, software development occurs within management structures and under pressure for resource efficiency. After software projects are specified to some degree, the required human resources are oftentimes estimated and allocated in so-called “man months”. This measure expresses the number of programmers required to develop a project within one month or how many months it takes one programmer to accomplish the project. This relationship between human resources and development time has been shown to be usually false (see Brooks [1995], pp. 13-26) due to differences in individual skill levels, for example, and the increased effort required for team co-ordination. Various corporate strategies and paradigms for managing software development and its resources have been proposed and applied, which are usually based on the sharing of “best practices”, formal definitions and specifications, on systematic breakdown
1. REVIEW OF LITERATURE AND WORK OF OTHERS

of projects into sub-projects and phase-based development processes (see Lecky-Thompson [2005]).

The academic and corporate software engineering fields use the term “design” frequently, but usually without further explanation. The issue of novelty generation is not treated as a key issue and instead of ideas and inspiration, the field usually cites requirements and specifications as the initiating drivers of projects (see, for example, Harrison et al. [2000]). The “software architecture” approach develops “the technical and methodological base for treating architectural design as an engineering discipline” (Garlan [2000], p. 261) and tackles problems arising from informal software design ideas by developing languages and tools for modelling projects in formal frameworks. Less formally oriented borrowings from the architectural design field include the sporadic acknowledgment of Rittel and Webber’s problem distinction as the one mentioned above, and the “pattern design approach” that was originally developed as a design method by Alexander et al. [1977] and later introduced to software engineering by Gamma et al. [1995]. This approach aims at exchanging and re-using expert knowledge within developer communities. This approach is further discussed in section 1.2.4.

Programmers use the term tool generally to describe utility programs that are
developed and used to accomplish routine tasks within computer environments such as shells, editors, parsers, compilers and debuggers, in contrast to software applications or operating systems (see Raymond [1996], p. 440). In “The new hacker’s dictionary” Raymond (ibid., pp. 449-450 and 459-460) describes “real hackers” as tool smiths who tackle “uninteresting” problems such as the development of user applications by generalising them sufficiently to make them interesting. The hacker’s dictionary admits that thereby “molehills are occasionally turned into mountains and mountains into tectonic plates” (Raymond [1996], p. 460).

In his account of the early days of “hacking”, Levi [1984], p. 126 points out that “an important corollary of hackerism states that no system or program is ever completed”. As an early insight gained by the “hacker” community, Levy (ibid. p. 127) notes that “[t]he problem of unrealistic software design is greatly diminished when the designer is the implementor”. Software developers, also being tool users, thus depend on strategies by which realistic tools can be supplied to them. Brooks [1995], p. 34 recommends that “[e]ach [software development] team will need its own toolsmith, regardless of the excellence and reliability of any centrally provided service, for his job is to see to the tools needed or wanted”.

There are parallels between the production and use of software tool use in software engineering and the production and use of physical tools such as scissors and screwdrivers. Software tools such as compilers or email clients are easily transferred from the contexts of their production to other contexts or users who can apply them to the intended problem class as often as they like. The challenge of determining problems at the time of toolmaking is encountered with a set of techniques that is virtually the same in different software development approaches such as generalisation and abstraction, modularity, re-releases of updated versions as well as thorough documentation and annotation of programming code to allow
future changes, possibly by others. The latter is commonplace in the open-source community, which collectively exchanges and develops programming source code in a virtually uncontrolled exchange that pays little attention to the distinction between developers and users. The open-source model is successful to the extent that it has a significant influence on software development on other, in particular corporate software development approaches. A widely known product of this community is the Linux operating system, a descendant of the UNIX operating system, which features a particularly wide range of relatively small tools. Raymond [1999] stresses the importance of involving communities of developers and users into the software development process, suggesting that “taming complexity” in open-source software development is a mere question of adding eye-balls (i.e. people, ibid., section 5 of the 2002 edition). He cites the text editor EMACS as an example of a robust and versatile tool whose development was (and continues to be) largely driven by user feedback (ibid., p. 37). Software developers in the early days of hacking in academic, corporate and open-source contexts seem to share a common appreciation for good ideas, elegant implementations and intelligent exploits. The origins of these are however rarely probed within these contexts. Raymond points out that “any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected” (ibid., p.54). Apart from stressing the importance of unrestricted community involvement in the development process, he offers no suggestions or guidelines as to how a programmer might proceed to achieve a truly great tool.

1.2.4 Design methods and design research

Based on the understanding that a method describes how something might or should be done in order to accomplish some goal, design tools are understood in
this thesis as “solidified aspects” of design methods. “Design methods research” and the “design methods movement” play prominent roles in the field of design research and can, more precisely, be regarded as a predecessor of what is referred to as “design research” today (see Cross [2007]). It must therefore be asked what the design of design tools can learn from the experiences gathered by the design methods field.

A number of sources review the history of the design research field including Broadbent [1979], Cross [1984] and [1993], Bayazit [2004] and Cross [2007]. There seems to be a general agreement that the field of “design methodology” and the so-called “design methods movement” came into existence and, for the first time, “received substantial academic recognition” in 1962 with the first “Conference on Design Methods” (see Jones and Thornley [1963] and Cross [1993], p. 63). The origins of design methods, however, date back well before the 1960s. Historical reviewers frequently emphasise that the new field was influenced by novel and pressing problems evoked by World War II and the so-called “Sputnik shock” (see for example Cross [2007], p. 1), that is, the Soviet Union’s successful launching of the first Earth satellites in late 1957 and this event’s repercussions in the Western world. Shortcomings in science education as well as in science funding in the U.S. were identified as some of the reasons for the initial failure to keep up with Soviet achievements (see Dickson [2001], p. 159, and Bayazit [2004], p. 18). Science and academic research were made out as keys in boosting technological advancement. At the same time they allowed emphasising the role of civilian research and down-playing of military involvement, which was of diplomatic interest in the U.S. This is reflected for example in the public’s reference to the engineers who developed the technologies required for the North American space programme (including Werner von Braun) as “rocket scientists” (see Dickson [2001], p. 253-254). With his “Man
on the moon” speech in 1961, President Kennedy set the (fixed) goal for NASA and the country “of landing a man on the moon and returning him safely to the earth” before the end of the decade. Much of the process by which this goal was achieved in 1969 took the form of systematically or “rationally” planned procedures, which are collectively referred to as the “systems approach”. The systems notion is used to describe objectively existing “parts of the whole and the way in which they behave in relation to each other” (see Hander [1970], p. 19). In this planning approach, the overall goal is usually taken as a given and divided into subgoals, sub-subgoals and so forth, which are then tackled individually, assuming that once all subgoals are reached, the overall goal is also accomplished1.

During the 1960s, design theorists took NASA as an inspiration to develop so-called “design methods” (see Rittel in Cross [1984], p. 318). These were defined as “the study of the principles, practices and procedures of design” (see Cross [1984], p. vii). “In some senses, there was a desire to ‘scientise’ design in the 1960s” (see Cross [2007], p. 1). Glanville recalls that in the early years of design research “Design should be Scientific. [...] Since Research should be Scientific, Design Research should be Scientific.” (see Glanville [1999] p. 80). The “World Design Science Decade” declared by Fuller for the period between 1965 and 1975 referred to in section 1.4.1 above also bears testimony to this scientific design research agenda. A widely known publication in the design methods field already mentioned in section 1.2.3 is “A pattern language: towns, buildings, construction” by Alexander et al. [1977]. The book is a catalogue of “patterns”

1An example from the early days of NASA is the method of stacking (existing) missiles and rockets on top of each other or bundling them in parallel to form multi-stage launch vehicles capable of reaching high altitudes and orbits (see Dickson [2001], pp. 38, 43).
of re-usable design knowledge, indications and counter-indications for their (re-) application and relationships between them. An example is “#196 Corner Doors” (ibid., p. 904 ff.), a recommendation to place doors near the ends of a room’s walls rather than at their centre so as not to disrupt circulation paths within the room. Alexander likens patterns to hypotheses of science (ibid., p. xv), essentially inviting their “refutation” by others and suggesting their replacement by more robust patterns. More formal examples of design methods usually took the shape of (potentially computable) procedures, that is, prescribed sequences of some kind of operations (see for example the “Design methods manual” by Cross and Roy [1975]). Alexander’s pattern approach was broadly dismissed along with other proposed design methods approaches in a rejection spearheaded by key proponents of the design methods movement themselves. Alexander proclaimed: “I would say forget it, forget the whole thing. Period. Until those people who talk about design methods are actually engaged in the problem of actually trying to create buildings, I wouldn’t give a penny for their efforts.” (see Cross [1984], pp. 312-313). Jones [1992], p. 73, another early proponent and later opponent of the design methods movement points out: “Methodology should not be a fixed track to a fixed destination, but a conversation about everything that could be made to happen. The language of the conversation must bridge the logical gap between past and future, but in doing so it should not limit the variety of possible futures that are discussed nor should it force the choice of a future that is unfree.”

Rittel and Webber [1973] reject the idea of designing by means of prescribed formal procedures on the basis of the nature of design problems. With a list of ten characteristics, they distinguish “wicked” design problems from “tame” non-design problems. Tame problems are characterised by the availability of described solution procedures as they exist in mathematics, board games, criminal law or
1. REVIEW OF LITERATURE AND WORK OF OTHERS

road traffic\(^2\). Wicked problems are described as inherently difficult to solve and the identification of their solutions as being implied in the formulation of the respective question, which is itself up to the designer to formulate. The simultaneous formation of design problem and design solution was formally observed and described by Dorst and Cross [2001], who propose the concept of “co-evolution” of problem and solution as an explanatory principle.

In what can be seen as a beginning disassociation of the design research field from positivist approaches to designing, Rittel, yet another initial participant and later deserter of the design methods movement, argues that a system “reflects someone’s understanding of something”\(^3\) (see Rittel [1992], p. 59). After systems had been regarded as being objectively existent in the world and able to be designed, controlled and evaluated as such, Rittel’s emphasis on subjective understanding is indicative of a paradigm shift on the pathway towards more recent radical-constructivist positions on designing and knowledge construction as those proposed by Glanville (see section 1.5.4 and appendix L). Dorst argues that relativist (subjective or “phenomenologist”) and positivist (objective) attitudes may both be appropriate at different stages of design processes, suggesting that a relativist attitude may be more apposite in earlier, novelty-generating and ill-structured stages while a positivist attitude may be more apposite in later stages during which design ideas are implemented and presented (see Dorst [2004]). Rittel also proposes a “conspiracy model” for design, acknowledging that designers, in contrast to members of other professions such as lawyers or physicians, do and

\(^2\)Experiences commonly made in road traffic demonstrate that the formal description of procedures and their applied practice can deviate from one another.  
\(^3\)“Ein System reflektiert jemandes Verständnis von etwas“. My translation from German.
should conspire with their clients (see Rittel in Cross [1984], pp. 326-327). Rittel thus argues for planning with others and hopes that planning for others or on behalf of others can be made obsolete (ibid., p. 326). He proposes greater emphasis of the argumentative process by which judgements are reached in designing (ibid., p. 323-324).

Design research today is less focused on providing prescriptive guidelines for designing. Frayling [1993/94], p. 5, Findeli [1999], p. 2 and Downton [2003], p. 2 categorise its lines of action into three subjects or modes of enquiry: Research for design, research about design and research through design. Findeli describes research for design as best exemplified by “R&D” – research and development work – in support of design. He describes research about design as “carried out under the heading of other disciplines (sociology, semiotics, economics, history, etc...) [on the subject] of design” (ibid, p. 2). Frayling describes research through design as “being achieved and communicated through the activities of [...] design” (see Frayling [1993/94], p. 5). Findeli relates the latter to what has also been referred to as “action research by project” or “project-grounded research” (see Findeli [1999], p. 2).

Despite their rejection by the field referred to as “design research” today, not all of the theories and approaches of the design methods movement and the systems approach have vanished. Some of them have gained footholds in other disciplines. Engineering design (as reviewed in section 1.2.2), which is largely based on the systems approach, developed in the 1980s (see Cross [2007], p. 2) and continues to exist as a parallel discipline to design research. It also forms much of the foundations of the academic software engineering approaches reviewed in section 1.2.3. Alexander’s pattern approach has, as mentioned in section 1.2.3, also been adapted in the software engineering field by Gamma et al. [1995], where it is used
1. REVIEW OF LITERATURE AND WORK OF OTHERS

to record and exchange expert programming knowledge\(^4\).

In his book “Understanding design”, Dorst’s [2003] presents a collection of short “reflections on being a designer”. In one of them, Dorst discusses the dilemma between realism and clarity that design researchers must face (ibid., p. 76). He argues that in trying to describe the reality of design practice in all its detail, one would be swamped in detail and that “any general statement (theory, model or method) about design must sacrifice some realism for the sake of clarity.” Dorst recommends the use of case studies and anecdotes from practice, which are richer and in some sense more complete. Dorst aims to solve the dilemma in his book by dividing design into small subjects, each of which reflects one facet of it, forming “a general but detailed picture of design, much like [a digital] image is built up out of many coloured pixels” (ibid., p. 4).

1.2.5 Design systems and CAAD research

In “The automated architect” Cross [1977], pp. 140 ff. discusses what a number of authors have referred to as “design systems”. Cross uses the word system as a super-category of products in the sense that a transportation system is a system within which a car is a product (ibid., p. 4). CAD systems are design systems in the sense that they can support the production of design outcomes. The term “design system” is here taken as synonymous with the term “digital design tool” used in this thesis. Cross (ibid., pp. 141 ff.) points out that the “enigmatic” nature of designing confronts the designer of a design system (i.e.

---

\(^4\)The observation that this approach of encoding methodological knowledge has failed in architectural design while it could succeed in computer science is directly related to the overall argument put forward in this thesis. The reflection offered in chapter 4 proposes an explanation for this.
the digital design toolmaker) with a difficult challenge when defining his or her objectives. As a consequence, “system designers” tend to seek firm ground from which objectives can be stated (for examples of lists of criteria previously proposed by others to guide the development of design systems, see appendix A). This results in a distinction of two types of operations observed in designing, which Cross (ibid., referring to Miller [1969], pp. 234 ff.) refers to as “magic” and “hackwork”, whereas magic describes the origination of design principles and hackwork describes the translation of design principles into actuality.

With digital tools appearing much more conducive to hackwork than to magic, Cross warns that this division would lead towards negligence of support for magic in design toolmaking. Previous reports on generative design tools suggest (or at least do not explicitly dismiss) their general applicability, which is often suggested by references to the tools’ anonymous “users” or “designers” in the third person. Examples include Frazer et al. [1999], Achten et al. [2000], Fischer and Fischer [2003b] and Chase [2005]. Intentionally or not, these accounts present generative software tools in analogy to tools like hammers, can-openers and bicycles in the sense that these tools are made by few but can be applied by many and that the users’ understandings of how the tools should be used matches those of the toolmakers. Ayrle [1991], p. 94 draws yet closer parallels between toolmaking for designing and for non-design tasks, suggesting the distinction between the user- and developer levels of computer tools for architects. According to Ayrle’s assessment of architectural education, most architecture students will

Miller’s choice of the term “magic” coincides with Heinz von Foersters lifelong interest in magic (see Poerksen [2003], pp. 4-5) and wonder (see Glanville [2003], pp. 100ff.) as key concepts underlying the second-order cybernetic thinking he developed and which forms the framework of the reflective analysis I present in chapter 4.
become software users who require some conceptual knowledge and a larger set of skills in the application of computer tools. Ayrle (ibid., p. 94) adds that some architecture students should be given the possibility to “step into software development and programming”. Ceccato [1999], p. 295 affirms that architects naturally make use of diverse tools as part of individual design processes. Accordingly, architects should be enabled to create their own computer tools as required, thus becoming “toolmakers” instead of mere tool users, indicating that open-ended explorative design processes require the designer’s tool kit to be open for additions by the designer as well as from elsewhere. Frazer [1995], p. 24 uses the term “toolbox” to describe a collection of concepts and generative processes for explorative conceptual design purposes. Frazer’s use of the toolbox analogy differs from the one usually used in software engineering (see section 1.2.3), where the primary concern of tool production and application relates to criteria of efficiency – or “efficacy”, a measure of a tool’s ability to produce a desired amount of a desired effect as conceived by the toolmaker. Observers who relate desired effects conceived by toolmakers to those demanded by tool users seem to tend towards sceptical assessments with regard to this ability. Contextualising his own work in developing generative design tools, Frazer [1995], p. 23 comments that commercial computer-aided design software “never quite seems to do what one wants in the way that one wants”, such that architects are driven to develop appropriate design tools by themselves.

Authors in the CAAD field distinguish between different modes and strategies of designing, which interrelate with different types of supporting tools. One such distinction is drawn by Kvan [2000] between “close-coupled” and “loose-coupled” relations between designers in the same project. At any stage of a close-coupled design process “an observer cannot identify a discrete contribution to the design
product by one designer or another” while in a loose-coupled design process “we see two or more experts operating in their own domains on a shared problem” (ibid., p. 411). Kvan argues that “[a] loose-coupled design process requires a very much different set of tools and conditions to be successful than a close-coupled one” (ibid., p. 415). Schmitt [1993], pp. 42-45 distinguishes between “top-down” and “bottom-up” oriented design approaches. In a top-down design process the designer proceeds towards a fixed goal by breaking it down into subordinate goals. Schmitt argues that approaches of this kind are ideally suited for computer implementation (ibid. p. 42). In a bottom-up oriented design process, overall solutions are achieved by the iterative combination of basic elements. Schmitt suggests that software tools can support this approach by performing sets of coded rules (ibid., p. 44). Recent mentions of this distinction include Chase [2005] and Scheurer [2005].

Aiming to offer “accounts of designer action, development of strategies for amplification of designer action in exploration and discovery of computational structures to support exploration” Woodbury and Burrow [2003], p. 517 describe the designerly process of novelty generation as the exploration of “design spaces” as illustrated in figure 1.3.

In this model of novelty generation, designing is modelled as networked points that wait to be discovered through exploratory design trajectories within design spaces. These spaces are assumed to be structured and designers find orientation in the structure to discover undiscovered space and designs. The network structure between the designs allows modelling design spaces and their contents as digital data structures. Describing their work, Woodbury and Burrow note that “[u]nderstanding design space trajectories privileges the easily explainable over the creatively explored” (ibid., p. 528).
1. REVIEW OF LITERATURE AND WORK OF OTHERS

Figure 1.3: Dimensions of design space accessibility, reproduced from Woodbury and Burrow [2003], p. 523.

1.2.6 Tools and media

Discussing the use of computers in architectural design, Glanville [1992], [1994a] distinguishes between design tools and design media. The two approaches are based on different understandings of the role of computers in the design process. Design tools are intended by their developers to speed up processes that would otherwise be performed by human designers, and assist users in executing predetermined tasks (Glanville [1992], p. 216). Design media, on the other hand, do not merely follow user instructions, but become active participants in design processes by suggesting alternatives to intended modes of action or thinking (ibid., p. 217). Applied as design media, computers can support design processes by giving users new, unintended and unexpected ideas.

Glanville’s distinction between design tools and media is based on the presence or absence of restricting control in conversations. This distinction is central to second-order cybernetic theory of designing, which is outlined in more detail in section 1.5.4. In the radical-constructivist view of second-order cybernetics, the process of novelty generation depends upon the construction of (necessarily subjective) meaning by individuals engaged in communication processes. Accord-
ing to this perspective, design processes can be described as conversations, with others or with oneself, typically via pencil and paper (ibid., p. 214). Glanville describes design not as a deterministic problem-solving activity, but as an open-ended explorative process: “In the process of this conversation, ideas grow and take on their own life, developing and changing in a manner that is essentially unpredictable and which increases in variety, richness and depth.” (ibid., p. 214) When engaging in such conversation processes, designers relinquish control over the design processes in order to arrive at novel outcomes.

Where this is not the case and where exchanges between users and computers are essentially restrictive, design tools are used in predetermined ways and hence are not conducive to novelty-generating design interaction. Current CAD systems for architectural design, which typically aim to provide designers with an “automated draughtsman” (see Glanville [1994b], p. 98), improve speed and efficiency but limit outcomes to the familiar. In order to provide support for design processes, Glanville [1994a], p. 5 calls for design software to accommodate and enhance the unpredictable nature of design processes by suggesting different approaches and changing attitudes in unexpected ways. Designers may enrich design processes through adapting tools as design media by what Glanville calls “constructive ways of abusing computers” (ibid., p. 6). He suggests a number of techniques to do so, seeking to evoke unpredictable and surprising outcomes. When using computers as media, designers are first of all advised to “listen” to computers to let them take part in design processes, remaining open to whatever they are offering. According to Glanville [1992], p. 221, designers may for example make use of interaction, distortion and collage, or provoke accidents and randomness to achieve unexpected but inspirational results. A detailed list of Glanville’s suggested techniques is given in appendix A.2. All approaches mentioned above
encourage designers to use computers to explore rather than to merely execute preconceived deterministic processes. As a result, computers become media for designing that are capable of amplifying creativity (see Glanville [1994a], p. 7).

1.3 Generative design and inspiration from Nature

This section presents inspirations that have motivated the objectives and strategies I followed at the earlier stages of the toolmaking study and which are based on analogies between Nature and design. It introduces generative design approaches that are based on natural and biological analogies and concludes with warnings regarding the limited powers of natural and biological analogies in designing.

1.3.1 Biological analogies in science and design

Natural analogies are used frequently in science, where explanatory principles are derived from observations of biological phenomena, oftentimes allowing or inspiring the generalisation of those principles beyond the scope of the initial observation. Examples include, as the titles suggest, Wiener’s [1961] “Cybernetics: or control and communication in the animal and the machine” and Holland’s [1975] “adaptation in natural and artificial systems”. Similarly, “Towards a theoretical biology” (see Waddington [1969]) investigates “whether we can develop a fairly general and widely applicable theory on the structure and dynamics of complex systems, which would be applicable in biology to work at the level of the population, the cell, development, and perhaps be of some use in the analysis of other complex systems of a social kind” (Levins in Ruse [1972], p. 106). In the same context others warn that biologists developing or applying reductionist models “are on the wrong track because they overlook biological complexity, biological order, and so on” (ibid., p. 106). Theoretical models derived from biological
phenomena such as organisation, adaptation, selection or complexity, oftentimes catch the attention of architects and designers, who are interested in achieving or relating their work to these phenomena. Work undertaken in the Artificial Life field (see Boden [1996]), which is regarded by some as the previously-sought theoretical biology (see Noble et al. [2000], p. 146) is of interest to many designers for this reason. The inspirations designers find in forms and processes of Nature are captured in Le Corbusier’s [1960], p. 155 proclamation “A plan arranges organs in order, thus creating an organism or organisms. [...] BIOLOGY! The great new word in architecture and planning.” Otto [1972] examines how the structures of bird skulls and spiders’ webs can inform the design of lightweight structures. Frazer uses the term autotectonics, to describe the “notion of seeding design through generic thinking, or designing acorns rather than trees” (see McLean [2006]). Zuse, being aware of von Neumann’s work in the area of self-reproducing cellular automata, gave a speech in 1957 (Zuse [1993], appendix 5) in which he envisions “technical gametes” which, if equipped with appropriate blueprints, could be used not to build, but to “plant” structures on the scale of factories and assembly hangars. Smith [1976] expresses a similar thought, discussing the possibility of buildings constructing themselves, “growing from a single brick-egg each” (ibid., p. 1). Jencks [1971], p. 47 predicted that “biomorphic architecture” and “growing structures” would appear in the 1990s.

1.3.2 Parametric tools for designing

The process of parametric design involves computer-aided form variation in analogy to Thompson’s observations of variations in animal form as shown in figure 1.4, in the sense that throughout parametric variation, similar to morphological variation between related species, shapes can remain topologically identical.
To illustrate the design potential offered by parametric modelling, Burry [1999], p. 81 introduces the term “paramorph” to describe parametric variations based on the same topological model. Paramorph 1 is based on an initially neutral box, which is progressively deformed by providing changing variables (see figure 1.5). While the initial box shape is rather neutral, parametric variations of the model result in more explicitly tectonic paramorphs that imply building structure or building volume.

According to Kolarevic [2003], parametric design changes the nature of the architectural design process: instead of designing the specific shape of a building, parametric design allows architects to design “a set of principles encoded as a sequence of parametric equations by which specific instances of the design can be
generated and varied in time as needed” (ibid., p. 18). Where in more static forms of computer-aided geometric modelling, model modifications require the changing of all relevant components individually to update the model, parametric design allows the propagation of changes in form of associated parameters throughout a model. By facilitating such changes, parametric design enables designers to develop a variety of related design options to choose from.

For the Swiss Re office building in London, the architects Foster and Partners used a parametric design approach to develop a 40-storey office tower that could suit a large set of constraints (see Munro [2004], p. 42). The overall ovoid form of the Swiss Re tower results in different floor plans at each level and a double-curved façade. Structure as well as building interiors follow a spiral order, with each circular floor plan rotated by five degrees relative to the previous one. The structure consists of a central core and a load-bearing façade of triangular elements generated by the intersections of inclined steel columns running diagonally and upwards along the façade in both directions (see figure 1.6). During the design process, parametric design was used to drive optimisation studies, and to rationalise structural components and details. A three-dimensional parametric model facilitated geometry analysis and allowed the design team to devise, test and choose from design variants. Despite the tower’s double-curved façade, this parametric optimisation process enabled the design
team to develop a variety of proposals that all maintained a cladding system consisting only of flat quadrilateral glass panels.

1.3.3 Evolutionary tools for designing

Genetic algorithms were popularised by Holland [1975] who presents their application in automated optimisation procedures in analogy to biological adaptation processes. In this approach, adaptation involves a progressive modification of what Holland calls “adaptive plans” that give rise to “structures” in response to given environments. In adaptive processes, possibilities for improved performance are usually exploited while the search for further improvements is still ongoing. In genetic algorithms, this progressive adaptation is driven by strategies modelled after biological mutation and genetic recombination, which enable algorithms to cope with large sets of interdependent parameters. Initially, information about which structures are most fit is assumed to be incomplete. To reduce uncertainty, the performance of different structures generated from the adaptive plan is tested within different environments. Genetic algorithms are developed as a response to tasks in which environmental conditions and related “performance measures” vary over time and space, such that advantages of given adaptations only apply at specific places and times. Such changing contexts, however, can be addressed only within the scope of initially defined parameters. This process produces families of related solutions that gradually develop towards the selection criteria defined in the fitness function. Alternatively to the use of a coded fitness function, generations of candidate solutions can be presented to human decision-makers for selection, rejection or further breeding.

Frazer [1995] pioneered evolutionary design in architecture as a “generating force for architectural form” (ibid., p. 9). He proposes to develop architecture
by expressing architectural concepts as “packets of seeds” and generative rules instead of static blueprints (ibid., p. 11). In the evolutionary architectural process envisioned by Frazer, architectural forms are developed based on code that contains instructions for form generation. Outcomes of this open-ended process are subsequently evaluated based on encoded selection criteria or by human designers who select appropriate or interesting candidates from automatically generated potential solutions. Evolutionary design is intended to automate explorative investigations by providing surprising and inspiring outcomes during the exploratory initial stages of the design process. Frazer’s approach represents a notable departure from previous approaches to computing in other fields, in which software performance is regarded as essentially deterministic. Figure 1.7 illustrates a conceptual study of an evolutionary design tool for the generation of wine glasses presented by Chan et al. [2001]. The system enables human designers to interact with automatically generated design proposals by either directly manipulating representations or by changing the rules, parameters and other control values of predefined evolutionary procedures offered by the software tool.

![Figure 1.7: Prototype of a generative design tool for wine glasses reproduced from Chan et al. [2001].](image)

While applications of genetic algorithms to design as described above are primarily intended to enrich divergent design processes by providing unantic-
ipated design proposals, evolutionary processes are also applied in convergent optimisation processes. Evolutionary structural optimisation (ESO) is based on finite element analysis and is applied to structural design tasks to optimise physical forms according to structural criteria. The optimisation process proceeds by iteratively removing inefficient material from a structural model or by adding material where needed to obtain the minimum volume under even stress-distribution. Each iteration of the automated optimisation process consists of finite element analysis to determine stresses, and the subsequent removal of inefficient material or structural elements from the structural model or the addition of material where needed (see Burry et al. [2005], p. 32). This process typically converges on a structurally efficient form of minimal volume. In a case study of Gaudi’s Passion Façade for the Sagrada Família church in Barcelona, Burry et al. (ibid.) demonstrate how evolutionary structural optimisation can be applied as a contributing generative architectural design tool in analogy to Gaudí’s hanging model (see figure 1.17), which also provide adaptive means for ensuring high degrees of structural efficiency. Figure 1.8 illustrates an evolutionary structural optimisation process for gravity and lateral loads that starts out with a rough initial volumetric configuration similar to the configuration found in Gaudí’s initial conceptual sketches of the Passion Façade. The outcome of the shown process arrives at an alternative structural solution to the existing structure, which is optimised primarily for gravitational loads.

![Figure 1.8: Evolutionary structural optimisation (from left to right) of the Passion Façade reproduced from Burry et al. [2005], p. 40.](image)

Evolutionary structural optimisation derives its name primarily from the itera-
tive nature of the process in which intermediate results are subjected to structural annealing rather than its similarity to biological evolutionary theory. Unlike the approach to evolutionary design taken by Frazer [1995], it does not employ a secondary genetic data structure which is manipulated by genetic operators in order to create the form of primary concern. Instead, it affects the primary form directly based on the structural performance of volumetric elements at the small scale into which it was broken down. Though evolutionary structural optimisation is a process that converges on a single solution, it may however be used as a generative design approach to obtain surprising results if the optimisation process is initiated with a relatively unconstrained initial structural model (Burry et al. [2005], p. 41).

1.3.4 Cellular automata tools for designing

In an early paper exploring the subject, Schrandt and Ulam [1970] propose a cellular automata system in two and then three dimensions that generates intricate spatial growth patterns of cells. In the three-dimensional variant, space is divided into a three-dimensional Cartesian grid, in which each position can be void or occupied with a cubic cell. From a given initial configuration, such as a single cell at position \{0, 0, 0\}, rule-based growth proceeds in discrete time intervals and creates successive “generations” of cells. Only occupied cells of the last generation are considered “alive” and are evaluated to give rise to new cells which are then added to the growing structure in order to be themselves evaluated in the subsequent execution cycle. Schrandt and Ulam (ibid., pp. 233-235) specify three rules according to which a new cell may be formed if:

1. it is contiguous to one and only one square of the current generation;
2. if touches no other previously occupied square except if the square should be its
1. REVIEW OF LITERATURE AND WORK OF OTHERS

“grandparent”;

3. of this set of prospective squares, of this \((n + 1)\) generation satisfying the previous condition, we eliminate all those that would touch each other. Again there is an exception for those squares that have the same parent; these are allowed to touch.

Lacking the computational facilities used today to implement and execute cellular automata, Schrandt and Ulam built a three-dimensional physical model of their cellular automata system described above, consisting of wooden cubes, which is shown in figure 1.9. The model shows \(1/24\) of the actual structure resulting after 30 consecutive applications of the growth rules listed above. The omitted parts are symmetrical repetitions of the shown part\(^6\). This structure illustrates how the recursive application of simple rules can generate intricate and unanticipated patterns. This capacity of cellular automata led to their application in the study of “complex systems” and (or including) design-related fields such as urban planning and architecture, where they are apparently appreciated in particular for their difficult-to-predict performance and their implicit spatial qualities.

Anzalone and Clarke [2003] and [2004] for example investigate cellular automata systems to generate non-uniform space grids, which they refer to as “differential

\(^6\)The shown part is repeated twice to fill one octant of the Cartesian coordinate system (assuming the initial single cube was placed at position \(\{0, 0, 0\}\)). This octant is mirrored into all remaining octants in a point-symmetrical fashion.
space frame systems”. They focus on differential space grids with non-uniform elements. These are related to cellular automata by exploiting similarities between the packing of CA and the packing structures underlying space grids, which are both based on discrete elements on spatial lattice grids. Differential space grids are generated in line-wise additive growth processes, which are driven by one-dimensional cellular automata systems. The cellular automata-generated patterns are then translated into three-dimensional form by mapping cellular automata cell states parametrically onto elements of the space grid structure (see figures 1.10 and 1.11).

Figure 1.10: Cellular automata as generators of space grid systems, reproduced from Anzalone and Clarke [2004], p. 156.

Anzalone and Clarke [2003], p. 328 aim to incorporate material and construction specific constraints into their cellular automata rules to control the interactions between large numbers of different strut elements in space grid systems. While Anzalone and Clarke [2004], p. 156 describe space grids generated using this process as interesting and potentially constructible, they also characterise the outcomes they yield as too limited in their formal range. Results generated with only few rules were perceived as too unpredictable and uncontrollable to be a valuable general design tool.

In response to the shortcomings observed in form generated exclusively through cellular automata execution, additional external manipulation mechanisms were
introduced to facilitate direct intervention by a designer, as well as adaptability in the generative process. Additional rules were used to keep the spacing between grid elements within structurally desirable ranges. A further set of rules was necessary to respond to contextual conditions and environmental constraints encountered in the context of applied design projects. Summarizing their results, Anzalone and Clarke [2004], p. 156) state that generative design processes require a balance of predictability and adaptability to be useful in design contexts.

1.3.5 Evolutionary Developmental Biology

Developmental biology is concerned with the processes by which single cells grow to form multi-cellular organisms. An interesting question regarding cellular development arises from its ability to express local and temporal characteristics based on one persistent set of instructions: How can different cells of an organism - which all share an identical set of genetic data - express locally and/or temporally different tissues and organs? What tells a muscle cell to differentiate into a muscle cell and not as a nerve cell though its DNA is identical with that of all other cells of its organism (including its nerve cells)? Arthur [1997], p. 103 explains: “Development is possible only if cells ‘know’ what to do […]. So the key question
becomes ‘how do they know?’, and the whole of developmental biology could be regarded as an attempt to answer this question.”

In order to develop from a single zygote to a multi-cellular organism by differential cell expression, individual cells use two basic strategies. Brenner, a pioneer in the field of developmental biology, designates them the “American” and the “European” way (according to Gehring [1998], p. 58). Cells developing in the American way often “may not even know” their ancestors and draw their sense of identity from interaction with their neighbors. The European way is for the cells “to do their own thing, not to talk to their neighbors very much” and to draw identity from their developmental history7.

In the field of modular robotics, Yim et al. [1997] proposed strategies that allow close-packing robotic modules of rhombo-dodecahedral shape to move over across other’s surfaces to approximate arbitrary three-dimensional shapes. This motion planning approach is based on sweeping a plane across a data structure corresponding to the target configuration, filling goal locations as the plane moves past (ibid., p. 5) and it can suggest parallels to the processes of cell crawling in higher organisms (see Stossel [1994]). Ingber furthermore suggests that a universal set of building rules (an “architecture of life”) might exist that guides the design of organic structures from simple carbon compounds to higher life forms, drawing analogies between tensegrity space frames and geodesic domes and the structure of organic cells and tissues (see Ingber [1998], pp. 48 ff.).

The fields of developmental and evolutionary biology have recently been merged into a unified field (see Goodman and Coughlin [2000]) dubbed “evo-devo” in recognition of their interrelatedness in the expression of biological form. While

7I have used both strategies in modelling two previously described case studies using the Zellkalkül environment. See appendix D for a report.
descriptions of genetic variation have been used as a paradigm for digital design toolmaking before (see section 1.3.3), models of cellular development have not yet been explored as a paradigm for digital design toolmaking, suggesting a potential “research gap” for experimental toolmaking studies.

1.3.6 Limitations of biological analogies

In “The evolution of designs”, Steadman [1979] scrutinises the way in which biological analogies are oftentimes used in science and design. Considering possible contributions of scientific research to design, Steadman warns that the application of scientific or rational thinking to design in order to make the design process itself scientific is misguided (ibid., p. 2). In his view, scientific study should be primarily concerned with the critical assessment of the products of design, but may also inform designers as part of a wider body of experience and knowledge. According to Steadman (ibid., p. 4), the main reason for designers’ interest in biology stems from the aspiration to ideas of wholeness, coherence and integration used to describe the organisation of parts within biological organisms. In design, biological analogy typically occurs in two interpretations, focusing either on visual appearance and composition or on aspects of function (ibid., p. 10). The first interpretation attributes beauty to wholeness, similar to organic coherence, and the second relates to the aesthetics of functionalism by attributing beauty to the useful or to expressions of usefulness (ibid., p. 9-10). Steadman argues that while biological analogies considered at a deeper level can be a source of understanding and of scientific insight, they are often drawn on in superficial ways. Biological and technical evolution, for example, are distinct processes that cannot be described in analogy to each other, as biological evolution does not involve teleological intentions (ibid., p. 186). Steadman (ibid., p. 168) further argues that biological
analogies are often derived not from the direct observation of Nature, but from codified generalised theories. This theoretical knowledge, however, is unlikely to support innovative thinking in the same way as the direct study of natural phenomena.

Ceccato [1999], p. 295 sees a fruitful relationship between natural inspiration and digital toolmaking for design: “The understanding of fundamental shape-forming processes in nature allows us to create tools that support our design intuition”. But he warns that this relationship yields no grounds for formal accountability or general validity beyond designers’ personal design concerns: “[p]rocess-driven design is not a science, and the use of tools reflects our own individuality in design. In other words, these tools add to the architect’s palette of instruments, which he can use as he pleases” (ibid., p. 295). Frazer [1995] warns that a distinction must be drawn between sources of inspiration and of explanation. Natural analogies used in science to explain or to illustrate depend on validity and correctness. Where natural analogies are used as inspiration and as take-off points for thought experiments, correctness and validity are less important, while misunderstood or even heretical ideas can provide much stimulation for the imagination (ibid., p. 12).

1.4 Space grid geometry

This section covers previous work by others related to the design of space grids. It comprises four sections: polyhedra and soap bubble geometry, a review of space grid structures and issues concerning their design, architectural geometry rationalisation and the different approaches towards rationalisation in design.
1. REVIEW OF LITERATURE AND WORK OF OTHERS

1.4.1 Polyhedra and soap bubble geometry

In dry liquid foam, faces between adjacent bubbles containing equal gas pressures, are flat. Dry liquid foam thus corresponds to space-filling packings of (usually irregular) convex polyhedra. Based on experimental observation, Plateau [1873] postulates laws (see Weaire [1999], pp. 24-26) describing the behaviour of soap films. These state that in a single, relaxed soap film, surface curvature is constant and that soap films meet in groups of three at angles of $120^\circ$ or $\cos^{-1}\frac{1}{2}$, forming an edge. In three-dimensional clusters of bubbles in liquid foam, edges meet in groups of four at angles of $109.47^\circ$ or $\cos^{-1}\frac{1}{3}$, forming a vertex. Based on experimental observation, Plateau also hypothesises that any contour, modelled for example by a closed wire loop, independent of how complicated its shape is, can be spanned by a soap film.

In 1887 Kelvin posed the question known as the “Kelvin Problem”, of how space can be divided into cells of equal size at a maximum cellular volume per surface ratio. Weaire [1996], p. 1 points out that this question has been the focus of much research attention. Kelvin notes that this problem is solved in foam (see Thompson [1887], reproduced in Weaire [1996], pp. 21-32). He obtained his own answer to the problem, known as the “Kelvin Conjecture”, from experimental observations of soap films. After dipping a cubic wire frame into soap liquid, Kelvin noted the formation of a stable configuration of soap films within the cube. From the eight

---

8 An edge between three soap films is also called a Plateau border.
9 Plateau’s laws describe stable foam configurations. Those configurations that do not meet the described conditions will change to stabilise and to meet the laws unless intermediate states during the process of change would require relatively larger total surface area (i.e. energy).
10 Regarding this so-called Plateau’s Problem see also Hildebrand [1987], p. 90 and Otto [1987], pp. 312-313
vertices of the wire frame, eight edges run inwards. They do not meet in one point but suspend a square soap film at the centre of the cube (see left of figure 1.12). Kelvin notes that the configuration around the square soap film resembles $\frac{1}{6}$ of a truncated octahedron$^{11}$.

There is an apparent contradiction in that the truncated octahedron has square faces and hence right angles between its edges while Plateau’s above-mentioned laws exclude the possibility of all but angles of $109.47^\circ$ between edges in three-dimensional soap films of minimal area. Close inspection of the soap films in the wire frame reveals that the edges are slightly curved to meet at $109.47^\circ$ (see also Thompson [1887] reproduced in Weaire [1996], figures 3 and 4). If this edge curvature is applied to the truncated octahedron, the resulting body, which is also known as the “Kelvin cell” fills space in close packing and was for more than a century the most efficient known structure in response to the “Kelvin problem”. The right of figure 1.12 shows a wire frame model of this structure built by Kelvin, which is sometimes referred to as “Kelvin’s bedspring”. At the centre of this model sits a white truncated octahedron surrounded by identical truncated octahedra in black in close packing arrangement.

In “On growth and form”, D’Arcy Thompson [1992]$^{12}$ gives a broad overview of natural forms and analyses their quantitative properties in a bid to encourage mathematicians to engage in and contribute to the fields of biology and morphology (see ibid., Prefatory Note)$^{13}$. Thompson argues that the diverse forms observed in Nature are determined by principles, such as surface tension, that can be modelled

---

$^{11}$This is a symmetrical, space-filling polyhedron with eight hexagonal and six square faces.

$^{12}$The original edition was first published in 1917.

$^{13}$This book has been an inspiration for much work in and reflection on design, including the interest in cellular growth and geometry underlying this study.
His overview includes discussions of aggregate natural structures, which the architect Frei Otto [1995] would later refer to as *pneus*, namely biological cells, bee’s cells and, to a lesser extent, soap bubbles. Recounting and illustrating various findings, including some made by Kelvin, Thompson investigates the “partitioning of space” (p. 555) and arrives at observations that are inherently related to tessellation issues that are also encountered in architectural space frame design and geometry rationalisation. Thompson, for example, describes how spheres made of a material allowing for plastic deformation (such as bread) can be piled in to a close-packed arrangement and then compressed to each take rhombo-dodecahedral shape filling space entirely (Thompson [1992], p. 526). This polyhedral shape, as Kieser [1815] noted and illustrated as early as 1815 (see left of figure 1.13), is a cellular shape commonly encountered in plant tissues. In later chapters Thompson discusses phenomena that can be related to more applied questions of form finding according to given design intentions such as structural efficiency (p. 958 ff.) and the theory of transformations (p. 1026 ff.) which presents the basic operations of what is referred to as parametric variation of form (compare Burry and Murray [1997]).

A mathematical approach for tessellating two-dimensional surfaces and three-
Figure 1.13: Rhombo-dodecahedral shape of plant cells as identified by Kieser [1815] (left) and octet truss as shown in Fuller [1975] (right).

dimensional volumes was presented by Voronoi [1907]. This approach begins with a set of points, which lie at the centre of cells into which the plane or the volume is divided by introducing dividing edges or planes between each two neighbouring points. Exploring Voronoi tessellations, Aranda and Lasch [2006], pp. 77 ff. draw analogies to natural formations in honeycombs, crystals and boulders.

In “Forming bubbles” the group around Otto [1987] presents its investigation of minimal, hence material-efficient and therefore light-weight architectural structures. Seeking to achieve the efficiency of minimal surfaces in built structures, Otto deploys soap films and photography as analogue computing devices for simulation and analysis purposes. The book presents Otto’s form finding experiments and results on minimal surfaces and bubble structures followed by a more theoretical summary of the mathematical models of minimal surfaces relevant to the group’s work. Apart from observation and largely visual and photographic analysis of soap film phenomena, this work transcends dominantly scientific, and hence analytically oriented previous work on soap films and ventures into designerly experimentation. While Plateau [1873], for example, was interested in the question of whether there is indeed at least one minimal surface for any closed three-dimensional curve, Otto is concerned with the various ways in which contours can be spanned by minimal surfaces and with the ways in which contouring structures
can relate to design intentions, purposes and other constraints and conditions such as gravity. The group also investigates the use of soap films to compute “minimal ways” (networks of paths of shortest total length) and the use of soap bubbles as “pneus” (see also Otto [1995]), individually, in clusters or in combination with other objects acting as structural elements. A well-known outcome of the group’s work on soap films is the Olympic Stadium in Munich with its minimal surface roof structure.

Several aspects of Fuller’s work interrelate spheres, polyhedra and space frames. His work on octet truss space frames\textsuperscript{14} (see Fuller [1975], pp. 135 ff.), is derived from closest-packed spheres as shown on the right of figure 1.13. His almost spherical “geodesic dome” structures such as the United States Pavilion at the Expo 67 in Montreal (see Baldwin [1996], pp. 166 ff.) are derived from polyhedra (see Kenner [2003], pp. 54 ff.) by the systematic introduction of new vertices and edges. Fuller described his work as “comprehensive anticipatory design science” (see Baldwin [1996], pp. 62 ff.) and he encouraged universities around the world to engage in his “World Design Science Decade” (see Baldwin [1996], pp. 63) between 1965 and 1975 to find ways to share the world’s resources more evenly amongst its human population. In response to wasteful and unsustainable design and resulting ecological and social problems, Fuller suggested anticipating and solving problems of the future before they become overwhelmingly acute. The design of energy and material efficient structures such as geodesic domes was part of that effort, which can therefore be said to represent an example of explicit and conscious geometry pre-rationalisation (see section 1.4.3).

As a resource to support Fuller’s “World Design Science Decade”, Critchlow’s

\textsuperscript{14}This type of space grid was originally pioneered by Alexander Graham Bell – see section 1.4.2.
“Order in space – a design source book” aims to show designers the basic “freedoms” of their constructions in space. He shows, in a thoroughly illustrated fashion, phenomena of spherical and polyhedral packing and different ways of tessellating space. In Critchlow’s view, the most basic shape to describe anything (including points) is the sphere, making spherically oriented analysis of space a necessity for planners who are “taught so soon to project [their] minds on to the ‘flat’” (ibid. p. 3).

Polyhedral configurations and space grids are effectively omnipresent in architectural design, usually however involving rectilinear and cubic or box-shaped configurations as in the Le Corbusier’s Venice Hospital (see Sarkis [2001]), Kurokawa’s Nakagin Capsule Tower (see Weaving [2004], pp. 53 ff.) in Tokyo and Kahn’s Washington University Library (see Ronner and Jhaveri [1987], pp. 94-97). In these cases, tessellations serve as ordering principles that relate parts to the whole within given compositions. The architectural use of polyhedra with less or more than six faces and three-dimensional tessellations involving angles other than $90^\circ$ are relatively rare. They include ornamental applications and surface decorations, vault structures in Gothic architecture as well as space frames. Polyhedra have also been used expressively at larger scales in Hecker’s Ramot Housing and in several projects by Goff (see Ristine in Gabriel [1997], pp. 35-126).

Almgren et al. [1976] propose a mathematical model explaining the geometry and behaviour of soap films and soap bubbles including the phenomena described by Plateau’s laws. The model draws on attractive molecular forces, which act to minimise overall energy by surface tension, thereby minimising surface area. Mathematician Brakke [1992] developed (and continues to develop) Surface Evolver, a software tool for the study of forms that are determined by surface-minimising tension such as three-dimensional soap films and foam. This software was applied
to experiments by various researchers, including Weaire and Phelan (see Phelan et al. [1995], who used the program in their study of foam structures (see ibid. p. 183) and thereby found a structure that satisfies the “Kelvin Problem” better than Kelvin’s own conjecture (see Weaire and Phelan [1994]). The new structure, called “Weaire-Phelan foam” has 0.3% less surface area than the Kelvin structure, whose surface-per-volume efficiency had been unsurpassed for 106 years. Figure 0.2 shows packed rhombic dodecahedra, a space grid resembling the Kelvin structure and a Weaire-Phelan-like space grid15. With its somewhat irregular polyhedral faces and containing two different polyhedra, the Weaire-Phelan-like structure appears visually more irregular.

In “The physics of foam” Weaire and Hutzler [1999] discuss the Weaire-Phelan foam within a broader treatment of the properties of various kinds of liquid and solid foams. With respect to the relationship between polyhedra and foam they point out the distinction between wet foams and dry foams. In “dry” foam, that is below liquid contents of 1% by volume, bubbles are separated by thin films, and provided equal gas pressure in adjacent bubbles, form approximate polyhedral shapes similar to the ones discussed above. With increasing liquid contents, soap bubbles in “wet” foam appear increasingly rounded and less polyhedral (see ibid. pp. 6-8).

The paper “Open problems in soap bubble geometry” edited by Sullivan and Morgan [1996] gives an overview of various open issues in the study of soap bubble shapes. Involving also issues of fluid drainage and structural performance, not all problems raised in the paper appear directly related to applied space grid design

15The grid in the centre and the grid on the right of figure 0.2 are different from the Kelvin structure and the Weaire-Phelan foam as they are composed of straight, not curved edges. Their vertex angles are hence not 109.47°.
as I examine in this thesis. The first item of the list, however, asks: “Is there any stable cluster of bubbles in $\mathbb{R}^3$ with some bubble being topologically a torus?”\textsuperscript{16} (ibid., p. 833). If stable here refers to a minimum energy state, then the sought type of cluster would likely contain only the edge and vertex angles described by Plateau’s laws while a three-dimensional torus topology implies that the cell in question replicates periodically along the mutually orthogonal x, y and z axes of the Cartesian coordinate system. In effect, what is asked for here is a space-filling lattice topology of a minimal set of vertex angles and repetitive edge lengths. Lattices with minimal sets of vertex angles and edge lengths are sought in the space grid design performed in this study. With respect to the question raised in Sullivan’s and Morgan’s paper, it is interesting to note that neither the existence of such a topology nor the in-principle possibility of it is known, nor methods appropriate for investigating the issue. This situation seems characteristic of the open questions in geometry that I encountered in the study presented in this thesis. It will be elaborated further in the discussion section 1.5.3 below.

1.4.2 Design of architectural space frame structures

Space frames were pioneered by Bell, who is commonly credited with the invention of the telephone (see Chilton [2000], pp. 1 and 168 and Makowski [1963], pp. 36-37). Bell developed and used the space grid geometry later popularised by Fuller as the “octet truss” (see right of figure 1.13) in a search for more efficient structures for kites. This structure is applied in numerous building designs including Biosphere 2 in the Sonoran Desert, Arizona by architects Margaret Augustine, Phil Hawes and John Allen (see Chilton [2000], pp. 88-89) and, on a much coarser scale, the Shimizu Corporation’s design for the gigantic TRY 2004 pyramid (ibid., pp. 160-163).

\textsuperscript{16}$\mathbb{R}^3$ denotes three-dimensional space.
1. REVIEW OF LITERATURE AND WORK OF OTHERS

The first large-scale architectural application of a glass-cladded iron space frame structure was the Crystal Palace for the Great Exhibition in London in 1851 (see Beaver [2001]), which showcased examples of the latest technology developed in the Industrial Revolution. For this project, self-taught architect Joseph Paxton drew on structural inventions made for his earlier design for an iron and glass greenhouse pavilion built to house the basin of a large Amazonian water lily. Paxton observed that the ribbed structure on the underside of the large water lily leaves was strong enough to support the weight of his young daughter, which inspired him to use ribs and cross-ribs in the structures of the lily pavilion and the Crystal Palace. Paxton’s design was an innovative response to the task of designing a temporary building at a very large scale: it had to be cheap, simple to put up, ready for immediate use and easy to remove (see Beaver [2001], p. 17). To cope with the size of the building and limited construction time – merely five months – the Crystal Palace made use of standardised, prefabricated parts that were mass-produced off-site and assembled on site. The structure consisted of iron columns, trusses and girders that formed a rationalised construction system that not only allowed the fast erection of the building, but also resulted in a previously unseen appearance.

Another innovative space grid system is the Reptiles system designed by Frazer. It is a lightweight enclosure system consisting of two basic folded plate units that can be joined together to form continuous structures (Frazer [1974], p. 232). Based on these two unit types, enclosures of a great variety of different shapes and sizes can be configured. The Reptiles system corresponds to the same sphere packing geometry as close-packed rhombic dodecahedra and forms the structural equivalent of what Fuller refers to as octet-truss space frames, without the frame (ibid., p. 232). The system was originally developed in 1966 as an enclosure for
a new gymnasium for Coventry Preparatory School that could mediate between several existing buildings of rectilinear forms, positioned at different angles relative to each other\textsuperscript{17}. Figure 1.14 shows the two basic unit types of the system (top left) and the proposed gymnasium enclosure (right). To cope with the potentially intricate forms produced by the \textit{Reptiles} system, Frazer developed a computer-aided design program that was initially intended to create drawings and perspectives of manually configured structures (ibid., p. 236). It was eventually developed into a program capable of proposing structures composed of basic units. Instead of a static data structure describing a desired shape, the automated design process starts out with a description of the structural units, the rules for combining the units, and a description of the “minimal closed object which the system can produce” (ibid., p. 236). This initial minimal object, the “seed” (see also Frazer [1995], pp. 68-73), is gradually adapted to the user’s requirements in a “cultivating” process consisting of repeated growing, stretching, deforming and pruning operations (see right of figure 1.14). According to Frazer [1974], p. 236, this cultivating process can involve human designers in dialogue with the system, or it can proceed automatically. The basic data structure is manipulated by coded functions, which retrieve, delete and update descriptions of the units, or insert units (ibid., p. 237). Once the data structure reaches a form that can satisfy the initially defined “brief”, it is translated into architectural form by deriving a description of the locations and types of units as well as their edges in Cartesian coordinates (ibid., p. 239).

\textsuperscript{17}Not entirely coincidentally, the \textit{Reptiles} system and the \textit{Life Game} cellular automata system were developed at the same time by Frazer and Conway, respectively, who shared the same computer system at Cambridge University (see Frazer [1995], p. 55).
1. REVIEW OF LITERATURE AND WORK OF OTHERS

Figure 1.14: Reptiles enclosure designed by John Frazer. Basic elements with variations (left) and top view in site context (right), reproduced from Frazer [1974], p. 234, 237.

When Chilton [2000] wrote “Space grid structures” during the outgoing 1990s, a relatively small number of visually irregular truss structures had actually been built, despite a number of designs having been proposed. Chilton discusses the visually irregular designs for a building extension at the Technical University in Lyngby by Robbin and Reitzel (see left of figure 1.15) and the Atlanta Pavilion by Scogin, Elam and Bray (see right of figure 1.15). While both structures remain unbuilt, a growing number of irregular space frames have been built recently.

Figure 1.15: Visually irregular truss structures, reproduced from Chilton [2000], pp. 119 and 167.

Federation Square is an urban-scale assembly of buildings serving civic, cultural and commercial activities in downtown Melbourne. Designed by Lab architecture studio in association with Bates Smart, Federation Square comprises nine buildings, all of which are clad in a façade grid that appears irregular and allows for variant
façade patterns. Each building can be identified by its individual cladding, which consists of different combinations of triangular panels of perforated and solid zinc, sandstone and glass. The façade grid common to all buildings is based on the pinwheel tiling, a triangle-based fractal tessellation of the two-dimensional plane. (see Lab architecture studio [2005], p. 46). The architects used the pinwheel grid pattern to generate visually irregular patterns for the building façades as well as for the atrium space frame, based on limited sets of elements. While the irregular façade patterns of Federation Square are two-dimensional, the atrium at the heart of the compound features a steel and glass structure derived from a three-dimensional interpretation of the pinwheel tiling. The structural system consists of two glass-clad envelopes, forming the inner and outer surfaces of the atrium, that are connected by a series of diagonal interconnecting members to form a three-dimensional space frame. Creating varying patterns of light, transparency and perceptions of depth depending on weather conditions, the glass-clad irregular space evokes comparisons to tree branches (ibid., p. 46).

Space frame design is also discussed in the engineering field. Borrego [1968] for example gives an extensive overview of grid topologies and their application as architectural space frames. “Steel space frames” by Ramaswamy et al. [2002] contains a chapter on “Structural design of space grid components” (ibid., pp. 21-48), followed by a chapter on “Preliminary design” (ibid., pp. 49-72). The prior reference makes extensive use of the word “design”, but mostly in a descriptive and analytical sense. It contains a case study of the design of a space grid to

---

18Lab architecture studio found the “pinwheel tiling”, on the Geometry Junkyard web site (http://www.ics.uci.edu/~eppstein/junkyard) as noted by Lab architecture studio co-founder Peter Davidson during a talk at the School of Design at The Hong Kong Polytechnic University on the 17th of November 2003.
support a concrete tennis court. As a case study, this section is retrospective, hence analytical. It does not contain prescriptive or instructional suggestions for the future design of comparable projects. Due to its function being that of a support platform, the design of the discussed space frame is not much concerned with architectural expression, spatial programming, circulation and similar aspects. Supporting a used space rather than covering it puts the project somewhat outside of the more usual range of space frame design projects. The reported “design brief” includes a rectangular layout plan with overall dimensions, six load parameters, a deflection constraint and tube specifications. The following section on “Design decisions” discusses the selection of a topology out of a choice of “the six grid arrangements normally used for space frames” due to its superior stiffness. It also reports decisions regarding the scaling of the topology, the structural height of the space frame and the spacing of purlins on top of it as the outcomes of seemingly separate considerations. While this orderly itemisation of design decisions may be seen as helpful in post-hoc explanation, their interrelatedness suggests they may have occurred in a more integrated fashion than is presented in the chapter.
Then follows a brief outline of the iterative, computer-aided structural analysis and optimisation procedure, by which the relationship between strut dimensions and dead weight was optimised. Starting parameters have been chosen “[b]ased on previous experience”. The concluding “Design summary” lists descriptive numerical parameters. Dilemmas, conflicting criteria or constraints, incomplete knowledge, collaboration within the design team, new ideas or indications of non-linear procedures (apart from the iterative optimisation) are not reported. The following chapter on “Preliminary design” of space frames (ibid., pp. 49-72) opens with the statement that “while analysis is a science, design is very much an art”. The authors proceed by pointing out the importance of “inputs from experienced experts, especially in the preliminary conceptual phases” (ibid., p. 49). The remaining part of the chapter discusses four “continuum analogy” based approaches for planning flat, regularly repetitive double-layer space frames. These approaches are described in the form of orthogonal line diagrams and presented along with geometric and structural analyses. The only suggestions of procedural guidance to the reader are concerned with the recursive problem of dimensioning structural elements\(^\text{19}\). None of the seven literature references that follow the chapter relate to the process of designing space frames.

The above-mentioned “Space grid structures” by Chilton [2000] contains a chapter titled “Design and construction” (ibid. pp. 52-60). Similarly to the majority of related publications, including the one by Ramaswamy et al., it presents

\(^{19}\text{Strut and node dimensions influence a space grid’s dead load while dead load, in turn, influences requirements for strut and node dimensioning. This recursive relationship can be resolved computationally, where possible, through iterative optimisation procedures. If a chosen combination of topology, scale, materials and detailing yields no structurally viable solution, these choices must be reconsidered, which is generally not taken care of computationally.}
1. REVIEW OF LITERATURE AND WORK OF OTHERS

solutions, constraints and regulations rather than guidance for designing. Issues of structural behaviour, thermal movement, seismic conditions and fire resistance are discussed along with orthogonal line diagrams of generic regular grid structures. Chilton limits procedural guidance to “Methods of erection” (ibid., pp. 56-58), listing the five most commonly used assembly techniques (ibid., p. 57) and giving practical advice such as the following: “In situations where it would be difficult to lift the whole space grid as one piece, or where it is not possible to assemble the whole grid on the ground, due to lack of space, the preassembly of units into a manageable area of space grid is a good compromise.” (ibid., p. 57). Comparable guidance for geometric enquiries and innovative space grid design is not given.

The publications that I have referenced in the engineering field describe known grid topologies and illustrate patterns of generic structures. Others, such as a chapter by Lalvani titled “Visual morphology of space labyrinths: A source for architecture and design” (in Gabriel [1997], pp. 409-426) or Kenner’s [2003] “Geodesic math and how to use it” describe methods for configuring structures of known kinds. They pay little attention to the processes of enquiry through which these structures have originated.

1.4.3 Architectural geometry rationalisation

In his design for the Casa Milà, Gaudí made the radical decision not to rationalise the organically curved form of the building, which might have allowed for more efficient manufacturing and assembly procedures. The construction work involved costly operations including the placing, removing and modifying of some stones up to four times. This strategy almost ruined the contractor who had agreed to finish the work for a fixed price (see Nonell [1980], p. 24). The design for the Casa Milà can thus be considered to lack “rationality”. The penalties involved in this deficiency
are tackled in a set of strategies commonly referred to as design rationalisation or geometry rationalisation. Burry [1993] and [2004] describes Gaudí’s use of ruled surfaces in the stone masonry for the Sagrada Família church geometry rationalisation, which not only serves the achievement of appealing shapes that acknowledge and accommodate structural necessities. It also serves the purposes of communicating complex building geometry to craftspeople and of identifying points and lines in space relevant for model making as well as for construction on site. Gaudí also used a hanging model to rationalise the structure of his design for the Church of Colònia Güell (see figure 1.17 and Otto [1989], pp. 19-20).

![Figure 1.17: Remodelling of Gaudí’s hanging model for the Church of Colònia Güell, reproduced from Otto [1989], pp. 83.](image)

The hanging model utilises the self-organising properties of hanging chains. These, when suspended from two points, form catenary curves, which produce structurally ideal load-bearing forms when turned upside-down. Since Gaudí, other architects including Candela, Catalano, Dieste, Gehry and more recently Cohen have used ruled surfaces as geometric principles in designing what Mitchell calls “extra-Euclidian” architecture (see Mitchell in Ragheb [2001], p.355).

In “The projective cast” Evans [1995] examines the relationship between architecture and geometry from a historical perspective. He counters the common perception that architects are mere consumers of geometry, rather than producers
of geometric innovation (p. xxvi) and that geometry therefore is used as a “dead” foundation for a “living” art (p. xxviii). Rather, Evans argues, geometry is not always dead (i.e. firmly established) at the time it is deployed by architecture (p. xxx). As Evans points out, the rationalising effect of ruled surfaces was not in all cases initially known or sought by those who used them after Gaudí.

A leading figure of the modern movement, Le Corbusier, in much of his career utilised planar surfaces to design rectilinear forms. Closer to the end of his career during the 1950s he developed an interest in “free” geometry and curvilinear form. This new formal vocabulary brought with it a need to find new strategies for rationalising complex geometries in order to achieve buildable designs. The roof geometry for the Ronchamp Chapel was apparently rationalised only after its design had been “finalised” to some extent by means of projective drawing (see Evans [1995], p. 301). Insights gained by making models of the design, however, seem to have been more crucial, in particular with respect to the process of producing construction data. Evans describes how modelling the roof by steel wire and paper implicitly rationalised it into ruled surfaces and how most of the building is defined by curved directrices and straight-line generators (Evans [1995], p. 302-303, 312-316). Evans explains that Le Corbusier had taken notice of Gaudí’s use of ruled surfaces for the roof of the Parochial School of Sagrada Familia as early as 1928 (see ibid., p. 333).

In his 1996 “Time calculated in seconds”, Treib [1996] examines the design and rationalisation of the extravagant Philips Pavilion in the second half of the 1950s. Le Corbusier was assisted by Xenakis, an engineer by training with an interest in music and architecture. Together with the Philips company, the building was conceived not as a product showroom but as a multimedia experience showcasing the potential of electronics in the arts (ibid., p. 11). Visitors would enter the building
at one end, experience a performance of about a quarter of an hour and then exit at the other end of the building. Le Corbusier conceptually related the purpose of the building to that of a bottle or of a stomach and initially put little emphasis on the pavilion’s exterior design (ibid., pp. 19-37). Xenakis had previously used ruled surfaces in musical composition (ibid., pp. 15-16). After Le Corbusier had begun to explore the stomach concept through sketching, Xenakis applied ruled surface geometry to rationalise the stomach design (see figure 1.18).

Figure 1.18: Iannis Xenakis’ sketch development for the rationalisation of the Philips Pavilion (reproduced from Treib [1996], p. 26).

As Treib explains, the result of Xenakis’ effort was not only an extraordinary building form with the structural capacity to support its own weight (ibid., pp. 59-63). Additionally, the intersections of generatrix and directrix lines produced a surface tessellation that allowed the prefabrication of concrete façade panels and the subsequent construction of the building (ibid., pp. 66-83). Both collaborators had reason to believe they had made the critical connection between the form of the pavilion and ruled surface geometry. The architect Le Corbusier originated the conceptual foundation of the design while engineer Xenakis rationalised it into buildable geometry. After a print article featuring the design for the Philips Pavilion cited Le Corbusier alone as the designer, Xenakis responded with a protest to Philips. Le Corbusier, dismissively referring to this objection as “l’incident Xenakis”, initially continued to claim sole authorship but a few days later agreed to the mentioning of both his and Xenakis’ names in design credits for the pavilion.
1. REVIEW OF LITERATURE AND WORK OF OTHERS

(see Treib [1996], pp. 85-89). This is an admission, which, as Evans points out (see Evans [1995], p. 296), no one else ever wrestled out of Le Corbusier. The relationship between the two collaborators, however, was permanently damaged and Xenakis was dismissed from Le Corbusier’s office soon thereafter (see Treib [1996], p. 88). As exemplified by Le Corbusier’s and Xenakis’ work on the Philips Pavilion, challenging projects began to require the architectural profession and the engineering profession to find ways to interact with one another.

In “An engineer imagines” Rice (see [1994], pp.58-66) recalls his involvement in the design and rationalisation of the Sydney Opera House from the structural engineer’s point of view. The Sydney Opera House with its iconic vaulted shells is frequently cited as the first challenging architectural landmark project that had to rely on the innovative collaboration between architect and structural engineer, which, in various modes of interaction, is practically commonplace in advanced practice today. It also had to rely on geometric rationalisation and the utilisation of modest computational power, which, after delays in the construction of the podium of the building, arrived just in time to perform necessary calculations on the self-supporting structure of the shells. The architect Jørn Utzon collaborated on the project with Arup engineers including company founder Ove Arup. After his originally envisioned parabolic geometry for the shells proved extremely challenging to construct, Utzon abandoned it in favour of a spherical surface.

While the Gothic master builder was a generalist responsible for the design of a building, its structure, its material logistics as well as the supreme leader of the workshop; the Renaissance period brought with it a more specialised role of the architect, who worked on a project together with artists such as goldsmiths, sculptors, painters and building craftsmen. The latter trade later specialised into those of the general contractor and of the professional engineer in England around 1850 (see Barrow [2001]).
geometry (see left of figure 1.19). With all shells sharing the same radius, they could be subdivided into individual elements, pre-manufactured and then assembled on site (see Mark [1990] pp. 3-5, Rice [1994], pp. 58-66, and Australian Government [2006]). As part of its argument for enlisting the *Sydney Opera House* in UNESCO’s list of World Heritage Sites, the Government of Australia cites the building’s role in fostering new methodologies as well as the innovative collaboration between architects and engineers following the separation of the two trades in the 19th century (see Australian Government [2006], p. 23, 35-42). While the outcome of this collaboration between Utzon and Arup is widely admired, Mark [1990], p. 5 cites Giedion [1967], pp. 676ff. who states that the building has also come to symbolise the tension between technology and art that still exists today. Evans [1995] describes the professional tensions that surrounded the project as a fiasco that “is rerun hundreds of times on a smaller scale” (ibid., p. 93).

**Figure 1.19**: Arup’s initial and final design schemes for the *Sydney Opera House*, Utzon’s sketches and spherical model of roof shells, reproduced from Australian Government [2006], pp. 29, 38, 39.

From this tension arises a question of authorship in much innovative architecture. Rice expresses regret for the lack of credit engineers receive. He argues “that there are many engineering contributions which go unrecognised, or which are attributed to the architects or others with whom the engineer is working” (see Rice [1994], p. 75). The tension between architect and engineer has been reported to be particularly fierce where the collaboration between the two professions gets closest and most successful. With respect to Arup engineer Cecil Balmond’s contributions
to the design and rationalisation of projects like Ito’s 2002 *Serpentine Pavilion* (see Balmond in Leach et al. [2004], pp. 128-135) and the *V&A Spiral* he designed with Libeskind, Jencks notes: “[t]o try and decide who did what gets you into an area of libel” (see Kabat [2001]).

A more recent example of geometry rationalisation is applied to the *Beijing National Swimming Centre* by PTW, Arup Sydney and the China State Construction and Engineering Corporation. The so-called “Water Cube” for the *Beijing National Swimming Centre*, which is currently under construction for the 2008 Olympic Games, was designed to resemble a transparent box filled with water bubbles (or, more precisely, with a dry foam). The irregular composition of bubbles envisioned for the design competition threatened to prove too complicated and too expensive to build. The solution developed by Arup engineer Carfrae uses a Weaire-Phelan based polyhedral packing as its geometric principle, exploiting its repetitive Cartesian make-up and irregular visual appearance (see Bosse [2004]).

![Figure 1.20: Beijing National Swimming Centre space frame. Top view rendering (left) and digital grid development tools (right).](image)

The packing structure shown on the right of figure 0.2 corresponds to the structure used in the Beijing project (see also sections E.3 and E.4 in appendix E), it is however not quite the actual Weaire-Phelan structure as is frequently reported. The Weaire-Phelan structure is the basis for this geometry, which is subsequently
“evolved” into a more tension-efficient derivation with single-curved edges - a system that would be more difficult to construct physically. In the case of the Beijing project, a large patch of this grid structure has been rotated before “cropping” it into its box shape. Therefore, while the truss structure shows no variation in bubble density, the box surface shows some variety in cell sizes and thus in visual density and visual irregularity (see figure 1.20).

Figure 1.21: Beijing National Swimming Centre grid structure development. Rotation and cropping of packing patch (left) and addition of planar facade grid (right).

Another recent example of a rationalised packing structure that is made up of only four polyhedra to assemble in a non-periodic close-packed structure is the derivation of the Danzer tiling by Aranda and Lasch shown in figure 1.22. From a space filled with with this structure, Aranda and Lasch subtract cells (or “boulders”) to achieve hollow cavities for Grotto, which is “elaborately artificial, absurdly fake” (see Aranda and Lasch [2006], p. 80). Marta and Grima [2005], p. 78 state that the “solution uses a combination of algorithms based on Voronoi geometries that transfer modularity from a Danzer tiling technique (developed by the Arup Advanced Geometry Unit) into a finite set of four faceted boulders. These four boulders fit together in a variety of ways, and the result is a wildly ordered three-dimensional pattern that never repeats itself.” The generation of this structure uses vertex points of the Danzer tiling as seed points for a three-dimensional Voronoi algorithm, that produces the four boulder types (see Aranda
Figure 1.22: Boulders derived from Danzer tiling (right) assembled to form the Grotto (left) by Aranda + Lasch, reproduced from Marta and Grima [2005], pp. 78-79 and Aranda and Lasch [2006], p. 89.

Software tools allowing the automated exploration of visually irregular grid structures appear to be rare. One example is CORDIN, a tool developed at Delft University of Technology by Huybers (see Parke [2002], vol. 1, pp. 449-458). The strategy underlying the tool is the translation of polygons in space to form polyhedra, which are subsequently replicated to fill space, following further optional translations such as rotations. The software, which appears to be based on experimental laboratory work rather then the result of applied designing, allows grid structures to be visualised as either polyhedral bodies or as space grids.

1.4.4 Pre-, post- and co-rationalisation

Otto [1987], pp. 11-12 notes in the context of his research into soap bubble structures: “In order to make his designs feasible the architect has the choice between two completely different methods of working: 1. The architect selects a shape which he considers to be aesthetically and functionally suitable and then chooses from the design available to him one which makes the conceived form feasible. [..] 2. the architect limits himself to well-known forms of construction which are certainly feasible and offer a solution as good as possible and thus
come closest to his vision of an overall optimisation. This method of using known optimal designs is the old traditional method of architecture. It was used to combine arches, vaults, beam and frame structures to form entire objects”. Loukissas [2003], p. 32-33 defines these two approaches more precisely: “In a rigorously post-rational system, the formal design is conceived in a process that is for the most part divorced from considerations about construction. A construction system is then retroactively imposed on the design. Certain compromises inevitably have to happen in order for the design to conform to any systematic means of construction. The opposite process is a pre-rational system in which the construction system is defined before the design process happens”. Loukissas presents this definition in reference to Whitehead of the Foster and Partners’ Specialist Modelling Group. Whitehead is credited with introducing the distinction between alternative rationalisation approaches in the context of computer-aided geometry rationalisation. During conferences and symposia, Whitehead has verbally referred to a third rationalisation approach as “embedded rationale”\(^{21}\). He describes this approach as integrating aspects of pre- and of post-rationalisation. The term co-rationalisation has also been used informally to describe approaches integrating pre- and postrationalisation, (see Fischer [2005]). When I asked Aish during a question-and-answer time following his keynote address on eCAADe 2004 in Copenhagen how irregular space frames could be designed to be made up of only few different types of components, Aish named three strategies, which he referred to as pre-rationalisation, post-rationalisation and in-process-rationalisation. The latter appears to be an alternative term to “co-rationalisation”, also reflecting Whitehead’s notion of “embedded rationale”. It generally seems to refer to an approach “between” or to a hybrid of pre- and post-\(^{21}\)See appendix K for an email by Whitehead explaining “embedded rationale”.
rationalisation with some emphasis on design process integration.

I have previously offered a preliminary definition of co-rationalisation in reference to Loukissas’ definitions, as a “process in which the compositional system is defined alongside and to some extent through the process of designing a form.” (see Fischer [2005], p. 2). A better description of co-rationalisation is still lacking, while previously reported design approaches suggest that pre- and post-rationalisation can be integrated in a variety of ways to give rise to combinations of both strategies. One example is Gaudí’s use of a hanging model to rationalise the structure for his design of the Church of Colònia Güell (see Otto [1989], pp. 19-20). The hanging model utilises the self-organising properties of chains, which, when suspended from two points, form catenary curves. When inverted, that is turned upside-down, the catenary curves describe optimal form for structures carrying loads in compression. While the designer is engaging in a hands-on form finding process with a hanging model, its physical components, hanging upside-down from the ceiling, at the same time find suitable solutions for the structural system of the building. This type of “embedded rationale” constrains possible instances of variable models to the set of desirable solutions and seems to relate co-rationalisation to the frequent and iterative adjustment typically observed in designing. One difference between the rationale embedded in the hanging model and adjustments in designing in general is that the principle by which the hanging model assumed structurally desirable shapes had to be known already before work with the model began.

Concepts underlying given geometry rationalisation approaches have been referred to by Glymph et al. [2002] as geometric principles. Glymph et al. use this term to describe the strategies chosen to break form down into elements that can be formally described and which relate to each other in some known way that permits
the construction of related forms. Discussing a parametric strategy for free-form glass structures using quadrilateral planar facets, Glymph et al. demonstrate how the translation of one curve along another one generates the facet vertices required to allow the post-rationalisation of a previously designed physical model shape for a double-curved roof surface (see figure 1.23).

The architects of Federation Square found a (previously known) tiling pattern and used it in their project. The engineers who rationalised the Beijing National Swimming Centre also utilised a (previously known) packing structure, which they found only after the overall architectural expression of the project had already been designed. The geometric principles underlying Ronchamp Chapel were identified through making a model of the design and the sphere-based principle that allowed the construction of the shells of the Sydney Opera House were only identified after the construction work for the project was well under way. Building on an existing tiling pattern, the designers of the Grotto have invented an entirely new tiling pattern for the project. These examples indicate that the time at which designers know geometric principles seems to relate only loosely to the timing of their commitment to them during given projects (that is, their choice of pre-, post- or co-rationalisation strategies).

1.5 Models of knowledge growth in science and design

This section reviews positions on knowledge production in science and design. It outlines changes in theories of science and in descriptions of the relationship
1. REVIEW OF LITERATURE AND WORK OF OTHERS

between science and design. It also presents descriptions of designing as an encoding activity at the border between coded and uncoded knowledge, and exemplifies the lack of formal methods for innovative enquiries in science and outlines second-order cybernetic theory of designing.

1.5.1 From objective truth to discursive growth of knowledge

As other practices, science is influenced by implicit beliefs and assumptions, which are oftentimes elemental to the extent that they do not enter the conscious reflection of those who are influenced by them. One such belief is the assumption that natural phenomena are governed by laws, which apply uniformly throughout space and time (see Kantorovich [1993], p. 59). Modern science is furthermore signified by the belief known as the Pythagorean outlook, which Galileo has expressed with the dictum that “the book of nature is written in the language of mathematics” (ibid., p. 59). Throughout much of the history of science, scientists have accordingly set out to “find” or to “discover” theories and laws that accumulate to produce formal and increasingly complete descriptions of what is observed in Nature. As such descriptions should not be subjective but equally true to everybody, Pearson noted in 1892 that “[t]he scientific man has above all things to strive at self-elimination in his judgments” (see Weizenbaum [1976], p. 25).

During the 20th century, the previously dominant view that science progresses by accumulation of new theory came under criticism. Popper, an influential philosopher of science of this time, instead describes scientific progress as a process of constant revolution, in which older theories are rejected and replaced by newer theories. In “The logic of scientific discovery”, he describes empirical falsifiability as the cornerstone of scientific theory, and states that scientific laws cannot be logically verified, but may be falsified based on empirical testing. According
to Popper, scientific theories that continue to resist attempts at falsification are not verified, but simply remain the best available theories until they are falsified and superseded by other, improved theories. In his view, scientific research aims to produce objective knowledge, with scientists searching and gradually approaching the truth through a process of proposing and testing hypotheses or conjectures. Popper characterises science as essentially critical, consisting of initial bold conjectures that are controlled by criticism (see Popper [1970], p. 55). Hypotheses or conjectures are meaningful only if they can potentially be falsified, and researchers should aim to refute rather than to confirm hypotheses in order to advance science. The testing of hypotheses takes the shape of rigorous empirical and quantitative evaluation. Popper described the progress of scientific knowledge in analogy to biological evolutionary processes, with competing scientific theories constantly subject to selection through rigorous attempts at their falsification. This critical feedback process of successive adjustments leads to gradual progress in scientific knowledge, towards greater problems and theories. Science is thus in constant evolution, and scientists partake in this endeavour by exercising critical rationality towards their own tentative theories and conjectures.

Popper’s view of the nature of scientific development was criticised by Kuhn [1970], who proposes that scientific development is not necessarily in constant evolution, as Popper describes it. Kuhn agrees with Popper’s position of the possibility and necessity of hypothesis falsification and the impossibility of conclusive hypothesis verification. Nevertheless, he disagrees with Popper’s position that scientists are only driven by logical criteria and will continuously attempt to falsify current theory. According to Kuhn (ibid., p. 6), the limitations of accepted theory are explored only under exceptional circumstances, in periods of crisis. In his view, most of what scientists do can be described as “normal science”, which
1. REVIEW OF LITERATURE AND WORK OF OTHERS

does not attempt to refute core hypotheses. Normal science can be described as “puzzle-solving research” that builds on accepted theory (ibid., p. 4). When engaging a normal research problem, scientists take current theory as a given, rather than challenging it. According to Kuhn, a field of science can only come into existence through the abandonment of critical discourse, which is revived only at critical points that require the fundamental challenging of accepted theory. Scientific progress is driven by psychological or sociological factors, and depends on scientific communities who do not challenge but maintain current theories. To emphasise this aspect of scientific progress, Kuhn uses the term “paradigm” when describing the tendency of scientists to sustain theories’ underlying value systems that are transmitted and enforced by institutions (ibid., p. 21). Only on rare occasions is accepted theory revised, which results in a “paradigm shift” and the establishment of new scientific paradigms. Popper [1969], pp. 56-57, in turn, insists that while scientists are not entirely neutral when assessing hypotheses, this does not prevent scientists from challenging fundamental assumptions. He rejects Kuhn’s suggestion to view scientific progress in terms of psychology and sociology, which he criticises as too dependent on “fashions and uncontrolled dogmas” (see Popper [1970], pp. 57-58).

Seeking to reconcile Kuhn’s and Popper’s viewpoints, Lakatos [1970], pp. 132 ff. proposes the concept of “research programmes”. In Lakatos’ view, scientific theories consist of several slightly different theories and approaches that share a common concept or “hard core”. Researchers involved in such research programmes typically attempt to shield this core from falsification attempts by providing additional “auxiliary hypotheses”, which they subject to modifications instead. Lakatos describes research programmes as being in either progressive or degenerative states: progressive research programmes will grow through
discovery of new facts or the development of new techniques or methods of prediction (ibid., p. 175). According to Lakatos, theories and research programmes can be assessed based on their progress: “We may appraise research programmes, even after their ‘elimination’, for their heuristic power: how many new facts did they produce, how great was their capacity to explain their refutations in the course of their growth?” (ibid., p.137) This assessment, however, can only be made in retrospect, as the future relevance of a theory cannot be predicted (ibid., p.173). Progressive research programmes adjust to challenge through change of auxiliary hypotheses, while degenerate research programmes lose this ability and can therefore be superseded by new research programmes. The vitality of research programmes as described by Lakatos depends on their ability to generate novel auxiliary hypotheses that can be tested empirically and reinforce current theory.

Popper states that scientific progress depends on “bold conjectures” and their critical appraisal (see Popper [1970], p. 55), but he does not offer an explanation as to how they are developed. Medawar [1979], p. 84 describes hypotheses as “imaginary preconceptions of what the truth might be”. They come from within and cannot be arrived at by using any known calculus of discovery. Only the subsequent testing of hypotheses can be logical and rigorous. According to Medawar, the truth is not in Nature waiting to be discovered, but is approached in a dialogue between the possible and the actual (ibid., p. 85), tentative guesses and empirical results. Popper’s description of scientific processes as recurring cycles of conjecture and refutation can also be characterised as a dialogue between two voices: the one imaginative and the other critical. To explain the difference between the domain of imaginative and informal sources of scientific ideas on the one hand and of rigorous and formal frameworks within which scientific ideas are tested on the other, Reichenbach distinguishes between the context of discovery and the context...
Kantorovich [1993], pp. 97 ff. notes that the philosophy of science has long ignored the issue of discovery. Treating the production of scientific novelty as a natural phenomenon, he proposes an evolutionary model of scientific discovery, in which novelty results from essentially “blind” serendipity and tinkering, the results of which are selectively retained. Kantorovich cites a range of serendipitous scientific discoveries, many of which are considered to be of outstanding importance (ibid., pp. 160 ff.), including Johann Kepler’s discovery of the laws of planetary motion, Max Planck’s discovery that light consists of discrete packets of energy known as photons as well as the discoveries of penicillin, X-rays, quantum mechanics and antimatter (see also Baggott [1990], p. 67). Kantorovich speculates what the origin of methods of scientific discovery might be, citing logical, epistemological, metaphysical and other possible influences on the scientist’s creative process (see Kantorovich [1993], pp. 58 ff.). He essentially describes the discovery process as based on serendipity (as illustrated in figure 1.24) and suggests that the processes by which scientific discoveries are blindly varied, selected and disseminated have social dimensions (ibid., pp. 197-199).

1.5.2 The relationship between design and science

Goel addresses the disparity between novelty-generating explorative thought processes and problem-solving processes in formal reasoning by examining how
the structure of human thinking relates to the structure of symbol systems (see Goel [1995], p. 9). According to Goel, thoughts of a certain nature require certain kinds of symbol systems, and the properties of symbol systems in turn affect the nature of thinking processes. Goel (ibid., pp. 9-10) criticises current cognitive theory, which focuses on well-defined and unambiguous symbol systems, and excludes thought processes that are vague, fluid, ambiguous and amorphous. For this reason, current cognitive science can only account for puzzle solving processes in well-structured domains (ibid., p. 218-219). Most problem-solving processes, however, involve a wider variety of symbol systems. Drawing on Nelson Goodman’s [1976] classification of symbol systems, Goel [1995], p. 166 focuses on three types of symbol systems: notational systems, discursive languages, and non-notational systems. Notational systems consist of well-defined and unambiguous symbols, for example ZIP codes or telephone numbers. Discursive languages include natural and artificial languages, such as English or calculus notations. Non-notational systems are exemplified by sculpture, seismograph readouts, or sketches.

Examining symbol systems used in design processes, Goel (ibid., p. 198) suggests that different symbol systems are related with different cognitive activities: Natural language, for example, is most prominent in the structuring of problems, while sketching is used to support open-ended and explorative cognitive processes during the creative development of design proposals (ibid., p. 218). Non-notational symbol systems such as sketches are conductive to novelty-generating design processes as they prevent early fixation or crystallisation via a dense ordering of syntactic and semantic elements and ambiguity of contents and/or referents. According to Goel (ibid., pp. 227-228), the process of mapping between notational systems, such as between analogue and digital notations, relies on negotiation of meaning and social agreement.
Wilden [1980], pp. 166-167 further argues that mapping discontinuity onto continuity is an epistemological necessity (see appendix M for a more detailed discussion of Wilden’s argument). He emphasises that most knowledge and understanding is analogue in nature, and that it is communicated analogically. In academic contexts, however, analogue knowledge is typically relegated to a marginal role. Analogue ways of learning, for example by imitation or simulation, are limited to art and music departments or medicine, where, Wilden argues, “no amount of digitalisation can properly describe the touch of a surgeon’s knife” (ibid., p. 167).

Criticising scientific approaches to epistemology as too limited by positivist technical rationality, Schön [1983] proposes an epistemology of practice that aims to recognise knowledge-producing processes better as they are observed to occur in practice. Schön specifically focuses on situations of uncertainty, instability, uniqueness and value conflict that differ from the well-formed problems typically addressed by rational, systematic and scientific approaches. Faced with such situations, practitioners tend to engage in what Schön calls “reflection-in-action”, an open-ended dialogue with the situation that does not separate between means and ends or thinking from doing (ibid., p. 68). Reflection-in-action proceeds through iterative experiments, which serve to “generate both a new understanding of the phenomena and a change in the situation” (ibid.). Reflection is triggered by the practitioner’s “surprise” at encountered situations (ibid., p. 280). In Schön’s view, professional knowledge lies less in technical expertise, but more in experience and more personal ways of knowing. Schön maintains that “the best practitioners know more than they are able to put into words” (ibid., p. 51). Such knowledge is different from formal knowledge in that it tends to be implicit in spontaneous, intuitive actions and qualitative judgements that Schön calls “knowing-in-action”
Knowledge is thus not necessarily separate from action to which it is applied in the form of rules or plans, but can be inherent in action.

Descriptions of the relationship of design and science underwent a transition over the course of the past forty-or-so years’ history of design research. Simon [1969], p. 132 argues for design to become a proper scientific discipline. Jones [1992], p.10 describes design as a hybrid of science, mathematics and the arts, while Cross [1982] describes design as co-existing next to the humanities and science. Glanville [1999], p. 89 states “scientific research is a subset of design, not the other way round”.

1.5.3 Lack of methods of enquiry

This section presents a review of four published reports on the methods of enquiry (or rather the lack thereof) through which geometric principles (such as those underlying the design of lattice configurations) come into existence in scientific practice. I have collected these reports in the process of identifying scientific sources of method in the context of my own methodological meandering outlined in section 3.5.

In 1952 there was limited knowledge about, and unsatisfactory models of, the molecular structure of DNA. After failing in a number of attempts to describe the structure, Crick and Watson ordered a physical modelling set to experiment with. Before the set arrived, Watson cut cardboard models of the four bases. He reports how from that point the Nobel-prize awarded discovery of the DNA double helix was almost immediate: “Suddenly I became aware that an adenine-thymine pair held together by two hydrogen bonds was identical in shape to a guanine-cytosine pair held together by at least two hydrogen bonds. All the hydrogen bonds seemed to form naturally; no fudging was required to make the two base pairs identical.
in shape” (see Watson [1969], p. 123). Once found, the double helix geometry accommodated what was known in principle about DNA before, including its replication capability. But without formal methods available for finding it, Crick and Watson depended on the testing of educated guesses on a trial-and-error basis.

**Figure 1.25:** Smalley’s paper model of $C_{60}$, reproduced from Baggot [1996], p. 69.

In 1985, in another geometric puzzle that would later be awarded with a Nobel prize, a team around Kroto, Smalley and Curl set out to explain the unexpectedly high quantity and stability of $C_{60}$ carbon molecules they obtained experimentally. The group developed a number of preliminary two-dimensional and three-dimensional hypotheses for what the molecular geometry might be, taking into account that the molecules’ observed stability may indicate a somehow closed shape. The search took the form of an intense debate for several days and involved stick models made of toothpicks and gummy bears, three-dimensional computer graphics as well as prior knowledge of Fuller’s geodesic domes. Making a cardboard model consisting of hexagonal carbon molecules, Smalley attempted to form flat configurations and then remembered a reference Kroto had previously made to a toy star-dome model formed of hexagons and pentagons. Expanding his experiments into the third dimension by including pentagons, he obtained the solution to the puzzle, a polygon with 60 vertices. Following the discovery, the team called their colleagues in the mathematics department to enquire whether
this geometry has been described before, only to be told that their model for C\textsubscript{60}, a truncated icosahedron, which they soon named *Buckminsterfullerene,* was geometrically identical to the pattern of an ordinary soccer ball (see Baggott [1996], pp. 48-76).

In 1961 Wang [1961] put forward a conjecture stating that tiles can only fill a plane in periodical arrangements, that is, in arrangements that are repetitive in two different directions. The conjecture was first refuted five years later by Berger’s [1966] proposal of an aperiodic tiling involving 20426 tiles. Following some reductions of Berger’s set of tiles, Robinson showed an aperiodic tiling based on six tiles in 1971, and in 1974 Penrose presented an aperiodic tiling based on only two coloured tiles. Penrose [1989], p.545 describes his discovery as having come to him intuitively “probably as a ‘flash’, but with only 60 per cent certainty!” Later, various formal methods (generative algorithms) for the automated production of Penrose tilings (see for example Rao et al. [2000]) were developed, showing that devising methods for producing known structures can be a relatively trivial task compared to the challenge of understanding a new and unknown structure. The open-ended and not formally structured search for an aperiodic tiling based on a single tile is still ongoing and, again, this search includes the search for searching principles.

As discussed in section 1.4.1, Lord Kelvin posed the question of what the most efficient surface-per volume tessellation of space into cells of equal volumes might be (see Thompson [1887], reproduced in Weaire [1996], pp. 21-32). The unsuccessful search for a more efficient structure lasted 106 years. Weaire and Phelan found one without actually searching for it (see in section 1.4.1). They encountered it accidentally while investigating the dependence of foam structure on the liquid fraction and related phenomena (see Phelan et al. [1995], p. 183).
Weaire describes the discovery as the result of “one third intuition, one third analogy and one third serendipidity [sic]” (see Weaire [1996], p.1). Discussing the newly found structure, he points out that there are neither means of knowing what the best solution to Kelvin’s problem might be, nor of identifying the Weaire-Phelan foam as the most efficient solution possible, nor of rigorously proving maximum possible efficiency of proposed structures. The same applies to the analogous problem in two dimensions. The honeycomb partitioning of an area into cells of equal size based on packed hexagons is the two-dimensional partitioning with the least required edge length known. But it is neither known what other area partition may require less edge length nor whether such a structure can in principle exist (see Weaire [1996], p. 1).

1.5.4 Second-order cybernetics

The description of design as “the human capacity to shape and make our environment in ways without precedent in nature, to serve our needs and give meaning to our lives” (Heskett [2002], p. 7) is consistent with the theory of homeostasis, which states that living organisms control their external environments so as to stabilise their internal environments. Similarly, Glanville [1997] describes goal-seeking behavior as aiming for stability in varying circumstances. This positions the study of design (and that of toolmaking for design) in the domain of cybernetics, the theory of communication, feedback and control as presented by Wiener [1961]. As an essential feedback and control system, Wiener (ibid. pp. 96-97) explains the function of a thermostat controlling a heater so as to stabilise the temperature of a room within certain boundaries. Below a certain temperature, the thermostat switches the heater on, above a certain (slightly higher) temperature the thermostat switches the heater off and so forth. Either side of the control loop have identical
numbers of states. Both can be either “on” or “off” and either of the two side’s states are mapped onto one of the other side’s states. This leaves no ambiguity in the system. This far, the account of the controlling thermostat and the controlled heater, as described by Wiener, describes a linear, cause-and-effect relationship that is at the core of what is now referred to as first-order cybernetics. More recently, second-order cybernetician Glanville [2000] points out that the heater and the warmth it produces control the thermostat just as much. The relationship is circular, and control resides between both sides. He (ibid., p. 4) points out that “Betweeness is the source of interaction and is also its mode and its site”. Any directionality (such as perceptions of intention or purpose) is attributed by an observer. Originator of second-order cybernetics Heinz von Foerster [2003], pp. 283 ff. insists that the observer is not independent of the observed system and must be considered part of it. He rejects the position that “[t]he properties of the observer shall not enter the description of his observations” as “a peculiar delusion within our Western tradition” (ibid., p. 285). Explaining the process of novelty generation, Foerster (ibid., pp. 273-282) argues (in reference to Gregory Bateson) that Newton did not discover gravity but that he invented it. The reason is that Newton merely put into words and proposed his (guessed) explanatory principle for a phenomenon, which, in its immediate presence, was “discovered” by probably every single person before and after the formulation of his law of universal gravitation. Taking the observer into consideration when considering systems, second-order cybernetics recognises four different relationships between systems and observers. The observer can be outside or inside the system and (s)he can look outwards or inwards, allowing for four different possible relationships between systems and observers. By regarding something observable as either stable or as unstable or moving, the observer places him/herself outside or inside
of the control loop respectively (see appendix of Glanville [1997]). The condition under which designing can occur is when the designer is in the system, looking outward at an unstable and possibly changing target. Glanville [1996] points out that different observers necessarily observe differently but are nevertheless able to communicate as if dealing with the same observed thing. Glanville [1996] also notes that, as receiving a message is not the same as understanding it, the transmission of a message is not the transmission of the meaning. Meaning is constructed from the transmission by the receiver. Building on Pask’s Conversation Theory (see Scott [2001]), Glanville models novelty generation in design on the control loop, placing interacting designers at one or both ends of it (this also applies to one designer assuming both positions in different roles). With humans being capable of expressing and responding to vastly more states than just two, the conversation between designers involves ambiguity. In contrast to the control loop between thermostat and heater, design communication thus generates uncertainty and ambiguity, which is essential in making new meaning and in designing (see Glanville [2000]). Glanville describes exchange loops, depending on whether they involve ambiguity or not as either “restricting” or “out of control”. Accordingly, Glanville [1992] classifies types of computer support for designing as either (restricting) “tools” or (out of control) “media”. Glanville describes [1997] second-order cybernetics as more general than first-order cybernetics, hence as more powerful and the only one needed. He likens the relationship of the two to a track (first-order cybernetics) left by a wheel (second-order cybernetics). Despite its simplistic and technical origins in first-order cybernetics, Glanville [1995], p. 5 notes that the view of second-order cybernetics “enriches us. It is deeply human, deeply humane”.

100
1.6 Summary and research questions

In this review of previous work of others, I have examined positions on tool use, on generative design, space grid and architectural space frame design as well as the production of knowledge in science and in design.

The use of tools by humans and other animals is examined from two perspectives. In the earlier perspective tools are of primary interest as utilities that allow exercising control over some part of the environment in a cause-and-effect manner. In the second and more recent perspective, the circular relationship between users and tools is of interest and the insights this relationship can offer to observers of tool use. The engineering design field develops, applies and evaluates tools based on criteria such as efficiency and safety. The origin of new ideas, either for new tools or in applying existing tools, is not of much concern. This approach is reflected in “managed” academic and corporate software development contexts. Unrestricted software development in the open source community also shows little concern for sources of novelty, but blurring the lines between users and developers, it supplies software that is appreciated for matching its needs closely.

Various design methods have been proposed, only to be later rejected and, the design methods field was deserted by its own proponents as inappropriately proposing “fixed tracks” to “fixed destinations”. Design research, a descendant from the design methods field, embraces three modes of enquiry: research for design, research about design and research through design. Some in the computer-aided architectural design research field, which seems to be influenced simultaneously by both the design field and the software engineering field, draw a clear distinction between developers and users of digital design tools while others proclaim the necessity of the architect being his or her own toolmaker. It is hence unclear whether digital design tools can, similarly to tools like scissors and
bicycles, be produced in one context and then transferred to and applied by others in other contexts. The generative design field has originated a distinct family of software tools, which, in contrast to other software tools, support non-deterministic processes. Often these tools use natural or biological analogies such as genetic variation which, in responding to formally expressed fitness functions, support convergent optimisation processes, and where applied in dialogues with users, can generate divergent processes of exploration. In the field of biology, processes of genetic variation and of cellular development are described as interrelated, which suggests the possibility of a new paradigm for a new family of non-deterministic design tools. Cautioning voices, however, warn that biological inspiration may be more appropriate as a creative inspiration and less appropriate as a key to scientific insight of general validity.

Space grid design has so far largely focused on regular structures. However, an increasing number of advanced architectural projects include visually irregular space frames. Space frames are closely related to questions of space tessellation, the packing of spheres and the geometry of soap bubbles, all of which can in principle accommodate irregular structures as well as regular ones. While structural engineering tends towards the use of tried-and-tested solutions, architectural design ventures to identify and build novel types of space frames, such as for example ones that are visually irregular. This typically produces a contradiction between a requirement for the absence of visual repetition on the one hand and a requirement for repetition from the perspective of component manufacturing and assembly economics on the other. The reconciliation of this kind of contradiction is referred to as geometry rationalisation. It is observed to proceed in different modes and under different conditions. Designers can commit to geometric principles before overall architectural forms are expressed in so-called pre-rationalisation processes.
Designers can also identify geometric principles suitable for rationalising proposed architectural forms post-hoc in so-called post-rationalisation processes. Recently, it was suggested that pre-rational and post-rational reconciliation of contradicting requirements for architectural geometry can proceed in integrated and computer-aided ways. The commitment to geometric principles before, during or after the design of an architectural form seems to be only loosely connected to the time at which involved designers get to know or design those principles.

Design and science are both described as processes of knowledge generation. Descriptions of knowledge generation in science have changed from descriptions of processes that uncover laws that govern objectively existing truth to creative interactions in communities that are nurtured by “surprise” and “serendipity”. Case studies of geometric innovation and in the wider field of scientific practice seem to confirm this view while philosophers of science are criticised for neglecting the question of what the sources or methods of discovery might be. Positions regarding the relationship between design and science have changed from descriptions of design as a subset of science to science as a subset of design, while designing is considered as a knowledge-encoding activity at the border between uncoded and coded knowledge. To formal procedures as those described in science and software programming, the uncoded is alien while design allows bridging the divide between coded and uncoded knowledge. Various authors describe knowledge generation and designing as “argumentative” (Rittel), “discursive” (Goel), as a “dialogue” (Medawar, Schön) and as a “conversation” (Jones, Glanville). Glanville’s conversational model of designing, being essentially symmetrical, places the knowledge-constructing “self” at one side of the conversation, without seeking to describe objectively what is known to the “other” at the opposite end. Ambiguity in conversations between both sides is the source
of novelty. Glanville’s model thus offers a framework for designing (and research through designing). Despite its in-principle symmetry, it is used from the subjective perspective of one side without attempting to explain what is internal to the “other” side, thus allowing to relate the what is uncoded and potentially possible to what is formalised as encoded knowledge.

The primary research questions that I have derived from the work of others reviewed in this chapter, and which are investigated in the remaining part of this thesis, are:

1. Can natural or biological analogies guide the development of digital design tools?
2. Can the development of digital design tools be justified after the abandonment of design methods?
3. Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers can?
4. Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?

Secondary research questions are concerned with more specific issues of toolmaking for grid design and geometry rationalisation. They are mentioned as aspects of the research strategies outlined in chapter 2 and the respective investigations are discussed at each stage of the toolmaking study in chapter 3.
This chapter outlines the overall structure of and strategic approaches underlying my own research work presented in this thesis. This work consists of two parts, the first of which is an applied toolmaking study, in which I approach the practice of digital toolmaking for design as research-through-design. The second part is a reflective analysis, in which I take a research-about-design approach to reflect upon the toolmaking study theoretically and develop a personal model of novelty and knowledge generation in science and design.
2. RESEARCH STRATEGIES

2.1 Introduction

The research work presented in the following two chapters consists of two parts, the toolmaking study in chapter 3 and its reflective analysis in chapter 4. Both the toolmaking study and the reflective analysis aim at examining the possibilities, the conditions and the limitations of making digital tools for designing, as well as the question of how the process of designing can be supported by making digital tools. Chapter 3 addresses these issues at a practical level while chapter 4 addresses them at a theoretical level. The toolmaking study and the reflective analysis, hence, differ in their approaches. This chapter outlines the respective research strategies adopted in the two chapters in different sections. Section 2.2 describes the toolmaking study, which consists of ten subsequent stages. Section 2.3 describes the approach taken in the reflective analysis of the toolmaking study, which uses seven aspects of second-order cybernetics as a theoretical framework. Section 2.4 presents a summary of this chapter on research strategies.

Figure 2.1: Structure of the toolmaking study and the reflective analysis.
Figure 2.1 shows the overall structure of the work as presented in chapters 3 and 4. The ten “beads on a string” extending from the lower left-hand side to the upper right-hand side of figure 2.1 represent the toolmaking study discussed in chapter 3. The “beads” represent ten subsequent stages of the toolmaking study, whereas the arrow-headed “string” represents the temporal progression of the toolmaking study. It is *not* to be taken as representing direct causal relationships or a procedure that was previously planned in this sequence. At each of the ten stages, the toolmaking study aims at a number of specific, temporarily stable objectives, which are outlined below. As a consequence of their temporal succession, earlier stages and the insights they offer contributed informally to the choices of strategies at later stages.

The seven circles “orbiting” around the “beads on a string” represent selected aspects of second-order cybernetic theory discussed in the reflective analysis presented in chapter 4, which looks back at the toolmaking study from an external perspective (i.e. not immersed in applied designing), drawing analogies between seven aspects of second-order cybernetics and observations made in the toolmaking study. The numbers shown in figure 2.1 refer to the respective section numbers in chapters 3 and 4. The research strategies and objectives of the toolmaking study and those of the reflective analysis are distinctly different from one another. They are therefore discussed separately in the following two sections 2.2 and 2.3. The review of literature and work of others leads to the formulation of the four primary research questions listed at the end of chapter 1. These four primary research questions are addressed throughout all stages of the toolmaking study and by all aspects of the reflective analysis. The extent to which they are addressed at different stages varies, however, roughly following the order in which they are listed at the end of chapter 1:
2. RESEARCH STRATEGIES

Question 1 (Can natural or biological analogies guide the development of digital design tools?) is mostly addressed at the first three stages (3.2 to 3.4) of the toolmaking study and in the reflective analysis in chapter 4.

Question 2 (Can the development of digital design tools be justified after the abandonment of design methods?) is addressed throughout both chapters 3 and 4.

Question 3 (Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers can?) is mostly addressed at the last two stages (3.10 and 3.11) of the toolmaking study and in the reflective analysis in chapter 4.

Question 4 (Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?) is mostly addressed in the stage discussed in section 3.8 and in the reflective analysis in chapter 4.

Secondary research questions are addressed in the immediate context of individual stages of the toolmaking study and documented at the respective locations below in this chapter as well as throughout the presentation of the toolmaking study in chapter 3. The secondary research questions are those questions I pursued intentionally and practically during the toolmaking study, researching through design. It was only later that the theoretical significance of observations made in pursuit of the secondary research questions became apparent to me. At this later point, I expressed the primary research questions a posteriori, thus laying the foundation for the theoretical research about design conducted in the reflective analysis presented in chapter 4.

2.2 Toolmaking strategies

The toolmaking study presented in chapter 3 investigates the four research questions put forward at the end of chapter 1 through designing and applying digital
design tools. Common to all ten stages of the toolmaking study is the development of three-dimensional space tessellations (that can be translated into architectural space frame structures) and the question of how the production of innovative space tessellations can be supported with digital design tools.

Most of this study takes the form of a qualitative enquiry, with quantitative methods taken into consideration and adopted at some occasions. Furthermore, this study is driven by an interest in the conditions of digital tool design, which necessitates immediate involvement in applied design processes, both at the level of tool making as well as at the level of tool evaluation through tool use. As the work presented in this thesis involved myself as a designer and evaluator, I report my efforts and my learning from a first-person perspective. The methods, application contexts and evaluation processes used in this enquiry differ from those applied in the scientific laboratory. The greater part of this study does not follow conventional patterns of formal scientific research. A more conclusive explanation for this is at the same time one of the findings of this research, and is discussed in chapter 4. At this point, I shall limit the explanation to two relatively straight-forward reasons. The first reason is that digital design tools are developed in various contexts with a notable difference in the resulting success between those tools that are developed and evaluated in academic laboratory research and those tools developed in advanced architectural practices. Those who develop digital design tools in laboratory research in academic design and architecture departments, and who thus may by occupation be expected to be more familiar with formal modes of enquiry, tend to develop and evaluate design tools according to either theory-driven hypotheses or according to personal curiosity but have so far not shown great impact in applied designing.\(^1\) Those who develop

\(^1\)This is demonstrated by the case studies reviewed in section 1.3 as well as by the
digital design tools in advanced design practices, and who are professionally more interested in effective results than in formal evaluation, develop digital design tools that have shown significant impact in applied designing. This suggests that formal evaluation might not be critical in making successful tools for designing. The second reason is, as Lawson [1997], states, that the design process “takes place inside our heads” (ibid., pp. 39) and is thus notoriously difficult to handle in empirical research. This applies to the assessment of design tools at two levels. On the one hand, the outcomes of designing and of design tool application are by definition new and unique and thus difficult to assess qualitatively and impossible to measure and to compare quantitatively. On the other hand, the development of design tools is in itself a design task. Designing is both a condition as well as an objective of this enquiry, which poses a challenge to formal causal statements aiming to inform future digital toolmaking for design.

Rather than carrying out one or more controlled experiments, into which variation would be introduced so as to allow an objective observer to measure resulting effects, I chose to take the perspective of a toolmaker operating from various perspectives in various contexts, including those of laboratory research and of applied design. The toolmaking study can hence be described as “research through design”. A few exceptions include stages during which the investigation of the research questions was temporarily (also) framed as “research for design”.

---

2 This is demonstrated by the applied case studies reviewed in section 1.4.3 as well as by the case studies of advanced practices published in Kolarevic [2003], Leach et al. [2004] and Chaszmar [2006].

3 As it was described by Frayling [1993/94], p. 5, Findeli [1999], p. 2 and Downton [2003], p. 2 (see section 1.2.4).
According to the position put forward by Foerster [2003], p. 285 (see also section 1.5.4), I acknowledge that as an observer, I am not independent of my observations. In design contexts in particular, there seems to be little reason to aim for “objective” or generalisable research findings. Chapter 3 therefore presents a first-person account of this study of toolmaking through design. The following outline of the investigative strategies I chose at different stages of the toolmaking study in sections 2.2.1 through 2.2.10 includes brief mention of the applications of the strategies as well as of respective results. These sections do not follow the more conventional pattern of an outline of planned methods followed by subsequent reports on their execution and results in separate chapters. Since results obtained and experiences gathered at earlier stages of the toolmaking study influence strategic choices at subsequent stages, it is necessary to outline such choices made at later stages of the toolmaking study with references to observations and outcomes I obtained at earlier stages.

2.2.1 Cellular development (reported in section 3.2)

My work during the first stage was motivated and inspired by natural and biological analogies in design and in scientific research as those referred to in section 1.3.1. At the centre of this inspiration lies the analogy between modular architecture and cellular development in biology. Given previous reports on the application of genetic algorithms as well as of cellular automata to design (see sections 1.3.3 and 1.3.4), and given the recent merging of genetic and developmental biological research (see section 1.3.5), I identify cellular development as an inspiration for the development of a digital design tool. This results in the primary research question 1 (Can natural or biological analogies guide the development of digital
design tools?). This question is purely curiosity-driven and not stipulated by an applied design project. Rather, it is framed as research for design, hoping to support applied design projects of others in the future. I investigated it in the form of a collaboration with a programmer in a laboratory setting. Based on literature research in the field of developmental biology, I identified key functions of cellular development and translated them into scripting code functions in the tool (see appendix B for a detailed list of these functions and appendix C for a simple example script for the tool). The secondary research question at this “research for design” stage is whether the computer-based simulation of the processes described by developmental biology can support form finding in design. As part of the tool development and testing process, I implemented experimental form-generating scripts and modelled some known examples of biological cellular development using the tool (see appendix D for reports on two such exercises). I then evaluated the results of these exercises on the basis of their potential contribution to applied designing. The test applications and experiments carried out resulted in the identification of two key challenges: The coding of massively-parallel cellular performance from cells’ inside-out perspective proved more difficult than expected. Results obtained from our test applications offered little contribution to design beyond the immediate scope of the tool development effort itself. Difficulties with respect to generative processes resulted mainly from the inside-out perspective required to program individual cells. This resulted in limitations on achievable outcomes, which, in order to relate to requirements outside the toolmaking efforts, must be known exactly at the outset of applications. Furthermore, the exterior surfaces of generated forms were dominated by the fixed cellular geometry supported by the tool.
2.2.2 Teaching scripting (reported in section 3.3)

Once the tool was considered sufficiently robust for use by others, I applied it as a teaching resource in a short generative design class of seven times 90 minutes contact time with eight second-year design students. This stage is framed as research about design. Part of the educational objective was to introduce students to scripting and to code-driven form development. From the perspective of the toolmaking study, my main objective of this tool application was to obtain feedback from students using the tool. The experiences with the tool until that point suggested that the apparent failure of the tool in aiding in the production of usable design output was due to a “perspective problem”. The secondary research question at this stage was whether students, taking a fresh look at the tool, could contribute input for further steps in the toolmaking process or to applications of the tool that lead to usable design results. The students however succeeded in using the tool only to the level that my co-developer and I have previously succeeded. Even in the context of unstructured, open-ended experimentation, the students assessed outcomes as too deterministic and requested the incorporation of new code functions to achieve more varied results. Main points of criticism were the difficulty of assuming the inside-out perspective required for programming developmental processes, and the geometric crudeness of external shapes produced by the software, both of which were previously noted, as reported in section 3.2.

2.2.3 Subway construction (reported in section 3.4)

As far as previous tool development stages suggested, the tool did not contribute to open-ended and explorative design in the way I hoped for as it left effectively all design decisions to the user, who is required to express them when programming
processes of cellular development. I therefore searched for design applications that would better relate to the tool’s requirement for predefined processes. In a research-through-design approach, I identified such an application in the area of subway construction. The secondary research question at this stage was whether autonomous behaviour of robotic elements can express patterns of an underground traffic network. This research question emphasised planning and construction processes more than form and appearance. This choice was motivated by my assumptions that issues relating to structure and to external appearance (both of which the tool does not address satisfactorily) are less dominant underground, while tunneling work might benefit from autonomous expression of form. Typical patterns found in subway tunnel networks suggested the possibility of coded rules that could drive the cellular development of a tunnel system in analogy to some of the architectural scenarios that inspired the development of the tool (see section 1.3), in particular the work on motion planning in modular robots reported by Yim et al. [1997]. My attempt to achieve rule-driven cellular expressions of subway networks was constrained by hardware limitations (due to the large numbers of necessary cells) and, again, by the required inside-out programming perspective. In most of the tool applications (apart from the biological modelling exercises reported in appendix D), the tool application at this stage was hypothetical and speculative while my proposed design responses remained preliminary and sketchy. I presented the growth model at an international conference (see Fischer et al. [2003]), where the biologically-inspired inside-out oriented form-generation approach supported by the tool in response to a design problem was criticised. The critics however failed to express the essential reasoning behind their argument at a level of clarity that would have allowed me to understand the principal shortcomings of the tool.
2.2.4 Parametric cell geometry (reported in section 3.5)

At this stage I addressed the problem of the “jagged” exterior surfaces of forms generated with the tool by incorporating parametric geometry control functions into the supported cells, taking a research-through-design approach. Through discussions and literature research, I compiled a list of six candidate approaches and chose the one which promised to be the most flexible and most easy-to-use means for geometry control. No known geometric strategy was immediately available to implement this approach and at this stage I did not succeed in identifying an appropriate strategy. We presented the problem to an expert audience at an international conference on computational geometry (see Fischer and Fischer [2003a]), where the problem was acknowledged as an “interesting” one. Nevertheless, we obtained no concrete suggestions as to how the geometric problem could be addressed. For the first time during this study, the question of the origin of methods and the issue of creative discovery emerged. Beginning to work on this stage of the toolmaking study, the secondary research question was by which geometric principle parametric shape control could be incorporated into packings of rhombo-dodecahedral cells. Soon, however, this question gave way to the more general question of where geometric principles come from and through what kind of processes they originate. Literature research into how experts find geometric strategies did not lead to usable guidelines or methods. Rather, it led to post-hoc accounts of successful identifications of geometric strategies (these accounts are summarised in section 1.5.3). Unable to identify theories of the origin of geometric principles at this point, I reverted to the issue of parametric cell geometry and implemented another, less challenging candidate approach from the previously compiled list. This approach offered means not only for parametric control of cell shapes but also for deriving data structures for generating rapid-
prototyping output. Once implemented, the tool’s negative impact on the external shapes of its output were still present, if shifted to a somewhat smaller scale. Use of these new parametric functions further added to the challenges of scripting from the cellular inside-out perspective, rendering the tool yet more challenging to use.

2.2.5 Re-orientation towards foam-like grids (reported in section 3.6)

The parametric cellular grids supported by the tool could be represented as space grids, which, in turn, can be translated into architectural space frame structures. This suggested applicability to the geometric space frame design for the Beijing National Swimming Centre, which was ongoing at this stage of the toolmaking study. One challenge of this project was the reduction of repetition in “foam-like” space grids to achieve visual irregularity while at the same time maximising repetition to allow for economic batch production of space frame components. This objective of geometry rationalisation for visual irregularity based on small numbers of different building elements was addressed in a number of ways in all remaining stages of the toolmaking study. This stage marked the point of critical strategic changes in the toolmaking study. The work undertaken prior to this point was aimed at fixed goals that I chose by myself and addressed at a pace determined by myself, following my own curiosity. After this point, work was carried out in a less self-paced, sometimes hurried manner that aimed at goals determined by others in specific contexts and located beyond the scope of my immediate toolmaking interest. Until this stage of the toolmaking study, ideas were followed-up to a conclusion and led to some form of test or evaluation. Work undertaken after this point aimed to support less well-determined objectives in mostly collaborative, open-ended processes, and not all ideas were stringently followed up and evaluated. Instead, the ideas and tasks that appeared most important, most urgent or most promising
were prioritised. Some of them were subjected to (formal) evaluation, others were merely developed to stages that allow the presentation of concept proposals. Effectively marking an important step into the direction of applied designing, I decided at this stage to discontinue the development of the cellular development tool \textit{Zellkalkül}. Instead of applying a robust tool to enquiries prompted by my own personal curiosity, I proceeded to address applied design challenges with rough, but “good enough” tools and methods \(^4\). Researching through design, without the tool, I re-visited the previously used cellular geometry and investigate the secondary research question of whether a visually irregular spatial tessellation could be derived from the previously explored packing of rhombo-dodecahedral cells. I succeeded in developing a grid structure that offered a number of choices for variety generation. Numerical evaluation showed that this visually irregular structure consisted of a limited set of different types of components, but lacking visual similarity with liquid foam, it was not applicable to the Beijing project.

2.2.6 Self-supporting façade system (reported in section 3.7)

At this stage another design project offered itself to apply and to evaluate the tools and methods developed so far in an applied context. The project required a visually irregular, self-supporting façade system based on few different types of building components, which cast tree-like shadows. Based on the geometry developed during the preceding stage of the toolmaking study (which was itself informed by cellular geometry used and developed during earlier stages), and taking a research through design approach, I investigated the secondary research

\(^4\)After this point, the only tool I developed for robust applications by others is the soap bubble tool applied in the postgraduate design workshop (see sections 2.2.10 and 3.11), which constitutes research \textit{about} more than \textit{through} design.
question of whether the previously developed geometry rationalisation approach can be modified so as to give the desired shadow effect. My design proposal was based on folded sheet metal that incorporates topological characteristics of the previously developed space grid structures and hence supports a comparable geometry rationalisation effect. This stage of the toolmaking study was the one involving the least time and effort. My proposed system was developed quickly in a pragmatic and rough way and results were presented in a two-page document to engineers of the project, resulting in positive feedback. The work undertaken at this stage marked a departure from the investigative approach I took previously in as far as it aims to design for an ongoing, applied project, through a rough and pragmatic approach to software and geometry development and by seeking feedback from experts involved in the project. This experience resulted in a re-orientation of the strategies I deployed in the remaining stages of the toolmaking study.

2.2.7 Development of irregular space grids (reported in section 3.8)

Following a conversation at a conference during which I was introduced to the distinction between pre-, post- and co-rationalisation of geometry in designing (see section 1.4.4), I then aimed to use these concepts to inform my toolmaking efforts. The secondary research question I investigated at this stage was whether the approaches described as re- post and co-rationalisation offer design procedures to achieve rationalised space grid structures resembling liquid foam. This stage of the toolmaking study thus also addressed primary research question 4: Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools? Using sketching on paper as well as a sketch-like approach using three-dimensional modelling software in a research-
through-design approach, I developed a number of space tessellations, all of which I succeeded in classifying as pre- or post-rationalised only in retrospect. I do not however succeed to develop space tessellations by following these approaches as guiding principles. This casts doubts on my assumption that these categories could inform toolmakers for digital design in their efforts. At this stage, I considered the development of a quantitative approach for the assessment of visual irregularity, which, following some initial visual comparisons, I dismissed based on the observer- and perspective-dependence as well as on the relationship between geometric characteristics of space grids and visual appearance. I also considered the computer-based structural simulation of a new, pre-stressed grid type I developed, and found that due to the new and “unknown” nature of the structure, physical testing would be necessary to allow for computer-based structural analysis.

2.2.8 Exploring co-rationalisation (reported in section 3.9)

The previous stage of the toolmaking study had indicated that pre- and post-rationalisation are themselves descriptive post-rationalised concepts that do not seem to offer much potential when used as prescriptive guidelines in tool development. Inspired by Robert Aish’s description of co-rationalisation, Hugh Whitehead’s notion of embedded rationale and by Gaudi’s hanging model, I further explored co-rationalisation at this stage of the toolmaking study, directly addressing primary research question 4: Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?

A first attempt at making a co-rationalising design tool for visually irregular space frames resembling liquid foam based on mobile bubbles was unsuccessful, due to geometric over-constraining. The following attempt at making such a tool
was inspired by the way in which digital calculators reduce continuous ranges of potential solutions to allowed points. In this attempt, I also integrated digital tools made by others for purposes other than designing in my own toolmaking. Using a research-through-design approach, I selected digital tools from the field of computational geometry that support the production and analysis of convex hulls, Voronoi tessellations and minimal surfaces. These tools are powerful and applicable to a broad range of geometrical problems but their (mostly numerically oriented and command-line controlled) interfaces restrict their use by most non-mathematicians. The secondary research question I investigated at this stage was whether the integration of non-design tools as “building blocks” into design tools constitutes a viable approach to toolmaking for design. I therefore developed graphical user interface “wrappers” which integrated these tools and supported the productivity of my own work with them. While my toolmaking efforts had previously not integrated tools made by others, I now used and appropriated such tools in my own toolmaking for design. While this approach provided inspiration, it was not conductive to my specific purposes. Replacing tools made by others with my own algorithm, I then succeeded in developing a tool for generating visually irregular rationalised space grids.

Numerical evaluation of the number of strut types contained in these space grids indicated an effective rationalisation strategy. The approach was based on a regular space tessellation space, in which a “virtual pressure parameter” is assigned to each cell. The cell sizes then change according to pressure differences, which does not necessarily affect node angles while different strut lengths cluster into batch-producible types. This procedure is illustrated in figure 2.2 and underlies all further grid-generating design tools developed in the remaining stages of the toolmaking study.
Truss rationalisation for cost-efficient batch production

1. Sphere packing arrangement
2. Derivation of regular truss
3. Assignment of virtual pressures
4. Bubble scaling according to pressures

Struts of relatively few different lengths can be produced cost-efficiently in batches, as a defined set.

**Figure 2.2:** Truss rationalisation method for cost-efficient batch production.

After having previously evaluated the degree of rationalisation in space grids by counting the number of contained strut lengths, I also investigated the possibility of counting the number of different contained node elements at this stage. This was only partially successful, mainly due to issues of computational precision.

**2.2.9 Tsunami Memorial Design Competition (reported in section 3.10)**

With a usable method for producing three-dimensional liquid-foam like structures implemented in software, I then tested the software by applying it in an international design competition using a research through design approach. Collaborating with members of the design team of the Beijing National Swimming Centre, I worked on an entry to the Tsunami Memorial Design Competition in Thailand. My secondary research question at this stage was to observe if and how tool use as I anticipated it during the preceding toolmaking phase differs from actually using it in a realistic
context. The first discussion on the project was conducted in person, all further communication was exchanged via email. Responsibilities within the team were to some extent assigned during the initial project phase based on individual skills and experiences. My design tool was available to my collaborator but I remained the only one using it during the work on the project. Over the course of the project, my collaborator suggested a number of alternative visual representations of bubble structures, which I supported by incorporating new functionality into the tool. I gained insights by observing my own use of the tool, my simultaneous further development of the tool, differences in the approaches and strategies proposed and adopted by myself and by my collaborator as well as by relating our design results to other submissions to the competition and comments made by the competition jury on the winning entry.

2.2.10 Postgraduate design workshop (reported in section 3.11)

After work on the Tsunami Memorial Design Competition indicated further that those other than the toolmaker seem to resist using tools that aim at supporting design, the aim at this stage was to investigate specifically research question 3 (Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers can?). For this purpose, I extended the tool used for the design competition and improved its robustness, so as to allow the observation of its application by others from a research-about-design perspective. Together with a co-investigator I conducted a design workshop with seventeen postgraduate design students, observing the use of two (my co-investigator’s and mine) design tools. We adopted qualitative observation strategies (user observation during tool use, tutorials and presentations, qualitative questionnaire items, recording of field
notes, user self-observation records) as well as quantitative observation strategies (automatic logging of user’s tool interaction, quantitative questionnaire items). We identified the key toolmaking intentions underlying the two design tools, categorised them according to Schmitt’s [1993], pp. 42-45 distinction between top-down and bottom-up oriented computer-aided architectural design (see section 1.2.5), Kvan’s [2000] distinction between collaboration and cooperation (see section 1.2.5) and Glanville’s [1992], [1994a] distinction between tools and media for design (see section 1.2.6). Following the workshop we related our observations and the participants’ self-observations to our original intentions as toolmakers. At this stage, qualitative and open-ended observation led to unexpected but valuable insights. Quantitative measurement of pre-conceived experimental variables did not offer significant insights, prompting a critical assessment of formal scientific approaches to observation and analysis in research about design. Key insights gained through open-ended qualitative assessment shed light on issues of control and the making of meaning in design conversations, suggesting that toolmakers have little means to support designing in contexts other than their own by offering digital design tools.

2.3 Reflective strategies

Following the toolmaking study and based on the experiences I gathered, as well as the results and the knowledge I gained during its ten stages, the four research questions expressed at the end of chapter 1 seem to be answered on an intuitive “yes-or-no” basis. The outcomes of the toolmaking study suggest to me that answers to all four questions should be negative (with exceptions under some conditions in the case of questions one and two). This leaves open the issue of “why” the four answers are mostly negative, which calls for a reflective
analysis of the toolmaking study from an external perspective and within a theoretical framework that describes the generation of knowledge and of novelty. For this purpose, chapter 4 presents a reflection on the toolmaking study and develops a philosophical argument on the conditions under which design can occur and the limitations under which digital design toolmakers operate. This reflection uses second-order cybernetic theory as a framework. The choice of this framework can be explained with the position taken by Jones [1992], who calls for design support modelled on a conversation, the language of which “must bridge the logical gap between past and future, but in doing so it should not limit the variety of possible futures” (ibid., p. 73, see also section 1.2.4). The conversation theory embraced by second-order cybernetic theory (see section 1.5.4 and Scott [2001]) bridges this gap by modelling design conversation as essentially symmetrical, placing the subjective “self” with its knowledge at one end of the conversation, subjected to the potentially unknown offered by the other end of the conversation. The reflection offers a foundation, upon which I then developed a descriptive model of novelty generation in science and design, which is based on Glanville’s [1999] description of design as a conversation and which accommodates previously drawn distinctions between “magic and hackwork” (Cross [1977]), between invention and discovery (Förster [2003]), between contexts of discovery and contexts of justification (see Salmon [1970]), Wilden’s [1980] and Goel’s [1995] distinction between the coded and the uncoded (or the analogue or non-notational and the digital or the notational) as well as an explanation for the notion of co-rationalisation.
2.4 Summary

The work presented in this thesis consists of a toolmaking study reported in chapter 3 and a reflective analysis of the toolmaking study based on second-order cybernetic theory reported in chapter 4. The report of the toolmaking study comprises ten stages, with each stage aiming at specific, temporarily stable research and design intentions. The toolmaking study deploys (in some cases considers and rejects) a research-through-design position based on various strategic approaches depending on each stage’s respective objectives, including qualitative and quantitative, informal and formal approaches. While earlier stages of the toolmaking study are primarily curiosity-driven, later stages address challenges arising from applied design projects. Along with this transition, my own view of designing underwent a transition from a positivist to a relativist perspective as outlined in appendices L and M. Outcomes of the toolmaking study suggest negative answers to all four primary research questions expressed at the end of chapter 1 (with exceptions under some conditions in the case of questions one and two). These answers are examined from a research-about-design position in a qualitative, detailed manner from a perspective external to the immediate toolmaking efforts in chapter 4. Based on this examination, I proceed to develop and discuss my own descriptive model of novelty generation in science and design, which incorporates notions and distinctions previously proposed by others. The overall effort presented in this thesis thus develops from an attempt to develop a generally applicable prescriptive tool for novelty generation in design to the development of a personal, descriptive model of novelty and knowledge generation in science and design.
This chapter reports on a ten-stage applied toolmaking study in the context of computer-aided architectural space grid design. At each stage, I investigate a different secondary research question, some of which are motivated from within and some from outside of the study. While the essential design challenge addressed in this thesis, the tessellation of surfaces and volumes, is common to all stages, I work in a variety of contexts over the course of the toolmaking study. At some stages, I make tools for myself, at other stages I make tools for others. At some stages, my work is removed from applied design contexts, while at other stages, I work within the contexts of applied design projects. At some stages my research interest is in the outcomes of designing, while at other stages, my interest is in design processes. Each stage of the toolmaking study addresses, either directly or indirectly, at least one of the primary research questions expressed at the end of chapter 1.
3.1 Introduction

This chapter outlines the toolmaking study that represents the largest portion of the research undertaken in this investigation. It is divided into ten sections, each reporting a stage of my investigation of digital toolmaking for space grid design from a particular perspective, with a particular objective (these are referred to throughout as secondary research questions) and following a particular investigative strategy. As discussed in greater detail in the previous chapter, my overall mode of operation during the toolmaking study is that of “research through design”. With the exception of my initial “research for design” approach, which I abandoned after the third stage, I understood toolmaking for design as a generic research and development effort for as yet unknown projects. Another exception is the “research about design” approach adopted at the last stage of the toolmaking study, during which I observed designers using digital tools.

Insights gained during the toolmaking study are outlined in each of the ten sections and are contextualised and reflected upon within a theoretical framework based on second-order cybernetics in chapter 4. The overall purpose of the toolmaking study is to investigate the primary research questions put forward in section 1.6 based on the review of literature and work of others. The primary research questions are concerned with the conditions under which digital design toolmakers can support designing, with the possibility of making digital design tools in one context and applying them in others, and with finding possible explanations for those cases in which digital toolmaking for designing is observed to be less successful. Findings regarding the primary research questions are reported briefly in the context of each stage of the toolmaking study, and in the summary at the end of this chapter. Based on observations made during the toolmaking study, the primary research questions are addressed on a preliminary
3. TOOLMAKING STUDY

“yes-or-no” basis. They are subsequently reflected upon theoretically on a “why” basis in chapter 4, which takes a closer look at explanations for these preliminary answers.

Individual stages of the toolmaking study are primarily concerned with more immediate secondary research questions that guide the practical research-through-design processes. These secondary research questions are outlined together with the relevant primary research questions at the beginning of each of the sections 2.2.1 through 3.10. Only the last stage of the toolmaking study, reported in section 3.11, addresses a primary research question directly. Each stage of the toolmaking study gives an account of the toolmaking efforts undertaken as well as its respective context and concludes with a report of the findings resulting from the respective stage regarding primarily the secondary research questions. The summary at the end of this chapter then draws together conclusions regarding both secondary and primary research questions. Some observations made during the toolmaking study represent dead ends in respective design developments and do not directly address any of the primary or secondary research questions. They are nevertheless reported in chapter 3 because they offer valuable exemplifications that will be referred back to in the reflective analysis offered in chapter 4.

3.2 Cellular development

The secondary research question investigated at this stage of the toolmaking study is: Can the computer-based simulation of the processes described by developmental biology support form finding in design?

The primary research questions addressed by observations made during this stage of the toolmaking study are question 1: Can natural or biological analogies guide the development of digital design tools? and question 2: Can the develop-
ement of digital design tools be justified after the abandonment of design methods?

In the field of computer-aided design research, biologically inspired genetic algorithms have been applied as techniques for generating design variants in “evolutionary” design processes (see Frazer [1995] and section 1.3.3). In biology, developmental and evolutionary studies have recently been merged into a unified field (see Goodman and Coughlin [2000]) dubbed “evo-devo” in recognition of their interrelatedness in the expression of biological form. The lack of a developmental counterpart to genetic algorithm-based computer-aided design offered a starting point for investigating the possibility of toolmaking for a new design approach based on cellular development. Inspired by the way in which higher organisms are observed to express their tissues and organs from coded lineages of cells, I co-developed a new tool, which is loosely based on the cellular automata paradigm and allows the translation of user-coded digital zygotes into tissues of differentiated forms for design purposes (see also Fischer and Fischer [2003b] and appendices B, C, D and G). The tool, named Zellkalkül, was intended to allow the evaluation of the new method and to be passed on to others for application in design. It provides the user with a virtual space in which cellular tissues can develop from single initial zygotes. Cells in Zellkalkül pack closely

1Torben Fischer joined this development at an early stage, henceforth implementing most of the software according to my specifications. Torben discussed this project as the subject of his BSc thesis (see Fischer [2004]) at the University of Göttingen, Germany.

2This name was chosen in reference to Konrad Zuse’s [1972] programming language Plankalkül. Being interested in the relationship between computer-aided design and cellular expression of form, I was intrigued to find that the three men commonly credited for inventing the digital computer, John von Neuman, Alan Turing and Konrad Zuse, largely independently investigated issues of cellular expression of form (see von Neumann [1966], Turing [1952], and Zuse [1969]).
to form tissues and can be visually represented as packed spheres, rhombic
dodecahedra or as octet truss space grids. A code script interpreter associated with
each cell allows cells to be programmed as well as to program other cells to act as
autonomous computers performing functions modelled after behaviors observed
in natural cellular development such as splitting, moving and so forth.\(^3\)

Programming digital zygotes to develop purposefully into lineage patterns
requires descendant cells to establish a “sense of identity” to guide pattern
differentiation. Natural cells achieve this by referring either to their developmental
history or to their neighborhood relationships (see Gehring 1998, p. 56 and
appendix D). Notwithstanding knowledge of these principles, the programming of
intended developmental processes, effectively from the “inside-out” perspective of
digital cells, proved challenging to myself. Similar experiences have been reported
in the Artificial Life field.\(^4\)

As an early test application of the tool I re-generated the shape of Frazer’s [1974]
Reptiles enclosure, which is based on the same sphere packing as cell structures in
Zellkalkül and hence also allows interpretation as an octet-truss space grid as shown
in figure 3.1. Other test applications of the tool included the modelling of known
biological development processes, namely differentiation in the eye-disk of the
fruit fly and the development of nematode embryos (see appendix D). Producing

\(^3\)Appendix B gives an overview of the scripting functions implemented to simulate
cellular development.

\(^4\) The difficulty of programming from the cellular inside-out perspective is also
indicated by Fleischer and Barr 1993, p. 401; Deutsch and Dormann 2005, p.
79. There are however successful examples which do achieve simple static goals.
Following a conjecture by John Conway that the Game of Life cellular automata
system could produce infinitely growing structures, a team around William Gosper
has for instance designed such a system in the form of the so-called “glider gun”. 

130
Figure 3.1: Perspective view of enclosure structure based on Frazer's *Reptiles* design generated using Zellkalkül.

previously known structures, the outcomes of these exercises had little to do with novelty generation. Other, less deterministic exercises and tests undertaken as part of the tool development and debugging process resulted in interesting looking “complex” patterns (see figure 3.2) that, however, did not address specific applied design challenges.

Figure 3.2: Fractal structures generated using Zellkalkül.

The key difference between this tool and other cellular three-dimensional modelling tools such as for example *DDDoolz* by Achten et al. [2000] and Strehlke’s
3. TOOLMAKING STUDY

[2001] *xWorlds*, whose building elements are manipulated directly by designers from an outside-in perspective, is the “inside-out” perspective from which cells express form in *Zellkalkül* is, in my view. This difference emerges as the primary obstacle in expressing form in directed, context-sensitive and intentional ways. This obstacle can, in its essence, be exemplified by the inability of symmetrically developing tissues to break their symmetry by their own means. Discussing his cellular reaction diffusion models, Turing [1952] (and others after him) has raised this issue⁵, stating that the three-dimensional variants of his model “certainly cannot result in an organism such as a horse, which is not spherically symmetric” (ibid. p. 41). The co-ordinated breaking of symmetry in cellular patterns is usually explained by the influence of some kind of input from the environment that influences pattern formation, such as the Earth’s gravity (see for example Newman and Comper [1990]).

The *Reptiles*-inspired enclosure structure shown in figure 3.1 was generated by executing only one active cell that was programmed with a static data structure describing the entire grid structure. Compared to the difficulty of programming a developmental lineage from the inside-out-perspective, possibly instructing individual cells to define their own destination based on inter-cellular communication, this approach made it easy to produce the desired configuration. The sequenced placing of the other, passive cells was possible due to the externally defined, uniform coordinate system and the similarly externally defined and uniform way in which cellular vertices, faces, cells and so forth are addressed at the scripting level. These references offer themselves, similarly to gravity, from outside the cells. By instructing the digital zygote to reproduce a static data structure, I effectively

---

⁵An early report on rule-based cellular pattern formation that remains non-symmetry-breaking was given by Schrandt and Ulam [1970].
gave up my intention of achieving a developmental process and replaced it with an intention to copy a known form. This observation began raising questions regarding the tool’s underlying idea of form generation from the cells’ inside-out perspective. Would references from the external environment be necessary or would they violate the idea of “artificial” autonomous growth? What if external reference points were more elaborate, taking the shape of entire scripts as in the case of the single cell generating the remaining portion of the Reptiles enclosure? If such a distinction between simple reference points and elaborate code scripts were drawn, then where exactly should it be drawn? Based on which distinguishing criterion? And why should this distinction matter?

In hindsight, I believe the essential question at this point was whether this investigation was to proceed as a basic research project aiming to build theory of cellular development or as an applied research project aiming to support designing. I began taking steps in the applied and pragmatic direction but for a while kept exploring alternative strategies to identify possible applications of the tool.

With respect to the secondary research question of whether the computer-based simulation of the processes described by developmental biology can support form finding in design, observations made during this stage of the toolmaking study suggest that the computer-based simulation of the processes described by developmental biology do not offer useful guidelines for the development of digital tools for designing. Main reasons are the limitations of achieved output in terms of their “jagged” exteriors and the challenge of coding massively-parallel developmental lineages from cells’ inside-out perspective.
3.3 Teaching scripting

The secondary research question investigated at this stage of the toolmaking study is: Could students, taking a fresh look at Zellkalkül, contribute input for further steps in the toolmaking process or to applications of the tool that lead to usable design results?

The primary research questions addressed by observations made during this stage of the toolmaking study are question 1: Can natural or biological analogies guide the development of digital design tools? and question 2: Can the development of digital design tools be justified after the abandonment of design methods?

As an alternative strategy to identify possible applications of the tool, I applied Zellkalkül in introducing scripting to eight second-year design students at the School of Design at The Hong Kong Polytechnic University in the fall of 2002. This class involved a contact time of seven times 90 minutes plus individual tutorials in person and via email. The students had no prior computer programming skills. This educational application of the tool with its constrained geometrical capabilities was expected to be useful for quick experiments as it constrains formal exploration, thus offering a very “controllable” design space. The tool served as a means to teach basic coding skills and as an environment to experiment with the generation of patterns of different degrees of complexity in three dimensions. Making use of only a limited set of the tool’s capabilities, students requested additional features to be incorporated during the course of the subject, such as time and date functions to relate to, in order to allow for greater variation amongst resulting shapes. The rhombo-dodecahedral cellular packing geometry of Zellkalkül is based on close-packed spheres and is not entirely Cartesian, which posed some

---

6 This educational application of Zellkalkül is discussed in greater detail in Fischer [2002].
challenges to students’ three-dimensional thinking capabilities and to the visual readability of generated forms. A simpler and more comprehensible packing logic might have been more helpful in this context. Despite the possibility of executing multiple cellular code scripts in parallel, the students focused entirely on sequential programming, that is, on scenarios in which only a single cell (usually the initial zygote) executes its code to control larger sets of other, passive cells. This choice of an “outside-in” perspective effectively contradicts the intentions underlying the tool.

Although the students succeeded in generating unforeseen cellular patterns (see figure 3.3), they did not use the tool for the intention-driven generation of anything of design relevance outside of the tool, apparently due to the same limitations I encountered before.

With respect to the secondary research question of whether students, taking a fresh look at Zellkalkül, could contribute input for further steps in the toolmaking process or to applications of the tool that lead to usable design results, observations made during this stage of the toolmaking study indicate a negative answer. Students could only contribute limited input for further steps in the toolmaking

![Figure 3.3: Sample page of student notes.](image)

Figure 3.3: Sample page of student notes.
process and did not succeed in applying the tool in ways that addressed applied
design challenges. The reasons are very similar to the ones encountered at the
previous stage (see section 3.2). Exteriors of generated output are dominated by
their “jagged” exteriors and the task of coding massively-parallel developmental
lineages from cells’ inside-out perspective proved too challenging.

3.4 Subway construction

The secondary research question investigated at this stage of the toolmaking study
is: Can autonomous behaviour of robotic elements express organisational patterns
for an underground traffic network?

The primary research questions addressed by observations made during this
stage of the toolmaking study are question 1: Can natural or biological analogies
guide the development of digital design tools? and question 2: Can the develop-
ment of digital design tools be justified after the abandonment of design methods?

My next attempt to propose a use for the tool was to develop strategies for
massively-parallel and decentrally controlled automated construction (see Fischer
et al. [2003]). This approach was inspired by speculative proposals that have
previously been put forward in the context of architecture and biology (see
section 1.3.1 for examples). More recent inspiration offered itself in the fields of
automated self-assembly and motion planning, in which Yim et al. [1997] presented
a motion planning strategy for swarms of autonomous, self-assembling robots,
which happen to share the same rhombo-dodecahedral packing geometry as the
cells in Zellkalkül. This strategy is based on centralised planning and parallel robotic
motion. In reference to the underground growth of plant roots, I proposed that
cellular robotic units could “grow” structures such as subway systems without
the need for centralised control. The developmental process should not follow
a precisely defined blueprint. Rather, it should aim to fulfill given requirements in a variety of possible ways, in analogy to Frazer’s “packets of seeds” approach (see section 1.3.3). The simulation of scenarios of this kind was hampered by limitations of available computing power but much more so by apparent in-principle challenges. Programming of massively parallel maneuvers which are not controlled from a single external vantage point was much more easily envisioned than put into practice.

Figure 3.4: Simulation of cellular subway station and tunnel development using Zellkalkül.

The pragmatic solution chosen in response to this problem was similar to the approach previously used by the students mentioned above in section 3.3. I programmed a zygote with a data structure corresponding to the intended cellular compound, to simply place cells at their target locations as opposed to instructing them in less deterministic ways to autonomously find their way. My approach was comparable to the one taken by Yim et al. [1997] described in section 1.3.5. In the case of the motion planning approach for modular robots used by Yim et al. as well as in the case of my cellular developmental model, the cellular units do not find their target positions autonomously from their “subjective” inside-out views but depend on guidance supported from outside-in perspectives. In the case of my model, the result (see figure 3.4) was not a simulation of the envisioned process but
3. TOOLMAKING STUDY

an illustration of preconceived outcomes.

I presented the outcomes of this stage of the toolmaking study at the CAAD Futures conference in 2003 in Tainan, Taiwan (see Fischer et al. [2003]). Following my presentation, a lively debate erupted centering on the questions of how and if at all the presented approach could be applied to design and, at a more basic level, whether the described robotic processes could be implemented in principle or not. A striking aspect of this debate was that the commentators, experienced computer-aided architectural design researchers, on the one hand seemed to generally agree that my approach was fundamentally flawed and heading into a problematic direction. On the other hand, however, they seemed to be unable to express why and how exactly the approach was flawed. Following the conference, the exchange extended into an email conversation during which one of the commentators, Prof. Robert Woodbury from Simon Fraser University stated: “The closer you relate these concepts to living organisms it may be the more difficult it will be to relate to architecture.”

With respect to the secondary research question of whether autonomous behaviour of robotic elements can express patterns of an underground traffic network, observations made during this stage of the toolmaking study indicate a negative answer. At this point, others as well as myself found it difficult to point to a satisfactory explanation for this answer, but an explanation will be ventured in the reflective analysis in chapter 4.

3.5 Parametric cell geometry

The secondary research question investigated at this stage of the toolmaking study is: Which geometric principles allowing parametric shape control could be incorporated into packings of rhombo-dodecahedral cells?
Research objectives at this stage did not have an immediate impact on the primary research questions, but they extended the investigation of the previous toolmaking stages at a more detailed level.

As a third strategy to identify possible applications of Zellkalkül I now addressed the tool’s shortcomings with respect to its form-related geometrical capabilities. In Zellkalkül, the basic cellular units assemble into composite configurations that are, in terms of their exterior shape, largely constrained to the system’s packing geometry and thus produce jagged exterior surfaces. This renders the modelling of most intended shapes effectively impossible, and had previously been described by my students (see section 3.4 above) as an obstruction to applying the tool in designing. The focus of this study therefore moved towards a search for geometrical strategies to support more flexible, more detailed and less regularly structured representations of form (Fischer et al. [2003]). At the core of this objective lies a dilemma between the desire for less regular appearances and the need for regularly structured, coded control. Individual cells need the capacity to change their polyhedral shape while the number of each cell’s neighboring positions, as determined by the underlying grid structure, must remain constant to allow unambiguous scripting control. We identified the following potential approaches (Fischer et al. [2005], pp. 442-443) to support the generation of “free form” using associative geometry:

1. Using large numbers of cells at a high resolution to approximate smooth three-dimensional shapes.

2. “Skinning” of cellular assemblies using surface fitting algorithms. In this

This dilemma is at the focus of the investigation of geometry rationalisation strategies for cost-efficient space frame manufacturing and construction discussed in later parts of this thesis.
3. TOOLMAKING STUDY

approach a secondary, tight-fitting surface geometry is “wrapped over” jagged cellular tissues to achieve a smooth exterior.

3. “Skinning” using cellular attributes to control a separate surface geometry, for example by mapping cell-related data such as cellular states or identity numbers onto curve control-points of a separate free-form geometry.

4. Using cellular attributes as parametric data to manipulate a separate form that is of primary interest and defined as a variable associative geometry.

5. Parametric position control of cell polygon vertices. Cells remain closely packed and their centre points remain fixed at equal distances.

6. Parametric control of cell sizes. This approach involves distorting the lattice of cellular center points.

My initial choice from this list was approach number 6. The main reason for this choice was my dissatisfaction with the geometric mismatch between cellular growth supported by Zellkalkül and natural larval development in *Caenorhabditis Elegans* as had become apparent in our developmental modelling exercises (see appendix D). While cell sizes in *Zellkalkül* are static and tissues with growing numbers of cells must grow in size, the zygote of *Caenorhabditis Elegans* is essentially of the same size as the developed larva, since cells divide into increasingly smaller descendants during the developmental process. By this time the work of Kitano et al. [1998] came to our attention, which aims at capturing and re-modelling the developmental lineage of the nematode worm’s larval development as well as its geometric properties.

What was required was a geometrical method by which cells of variable sizes could remain close-packed in a three-dimensional configuration. Using this method, size changes of individual cells would also result in position and size changes of respective neighbouring cells, ensuring that the number of direct cel-
lular contacts remains constant. A possible result is illustrated two-dimensionally in figure 3.5. This method should allow cells to break their symmetry and to take less regular shapes if necessary while constraining cellular size variation in such a way that every cell’s number of neighbours remains constant at 12. This was important in order to maintain the logic by which cellular neighbours are addressed in Zellkalkül’s scripting language (see appendix B). Failing to identify a method that meets this requirement and failing at identifying the process by which methods of this kind are produced (so that we could use these processes to produce the required method), we decided to present the problem to experts in the field. We wrote a short report on our toolmaking effort and stated our problem in a paper (Fischer and Fischer [2003a]), which we presented at the European Workshop on Computational Geometry 2003 at the University of Bonn, Germany. The paper was accepted for presentation and received as “interesting” by the workshop participants. Some forty papers were presented in total, which, to our surprise, tended to present solutions in the form of procedural methods, while ignoring the source of the processes by which the authors arrived at the methods of computational geometry they presented. Following this experience, I decided to look at descriptions of what the methods of enquiry might be that lead to innovative geometrical procedures. This search has resulted in the collection of
the four case studies presented in section 1.5.3, which indicate that no such general method exists. I have later found a solution to the problem and applied it not only to packings of rhombic dodecahedra but also to packed truncated octahedra and a Weaire-Phelan-like structure (see appendix E) as discussed in greater detail in sections 3.9, 3.10 and 3.11. As will be described in section 3.9 and in footnote 24 in section 3.11, I did not arrive at the solution by applying a formal method or by rational reasoning alone but by considering many different options and drawing analogies that were not immediately obvious, such as analogies to balance scales and voltage division in electrical circuits.

For the time being, we accepted that the preferred strategy of parametric control of cell sizes was too difficult to achieve and had to consider an alternative strategy from the above list. We eliminated option 1 as it does not achieve but merely approximates smooth surfaces at the cost of exponentially increasing memory consumption during form generation. This strategy would not have required a technical solution since all that would have been needed to achieve it would have been the use of a significantly larger number of cells (which was practically prohibited by hardware limitations). We also eliminated options 3 and 4 as these two strategies contradict the basic intention behind the development of Zellkalkül, which was to provide a single unified geometric and data structure. We settled on option 5, which we believed offered a practical response to the above-mentioned dilemma between irregular visual appearance and the need for regularly structured, coded control without involving parametric mapping onto secondary geometric shapes or necessitating very large numbers of cells. In this approach, all faces of the rhombo-dodecahedral cell shapes are split into two triangles, allowing parametric movement of cellular vertex points as well as straight-forward data output in STL format for rapid prototyping purposes. As
a first step, the rhombic cell faces were triangulated by introducing a new edge between the two closer vertices of each rhombic face in order to allow vertices to move individually without affecting other vertex positions of the same face, while continuing to allow “tissues” to remain close-packed without void or overlapping spaces. This triangulation is also useful in facilitating the creation of triangle-based output file formats such as STL (as specified in Burns [1993]), which is necessary for producing stereo-lithographic rapid prototype models of generated form. For these two purposes, we split all the rhombic faces of the dodecahedral units into two triangles as illustrated by the dashed lines in the middle diagram of figure 3.6. As a next step, it was necessary to identify the movement ranges of all vertices in order to achieve useful geometric constraints for parametric manipulation. By “movement ranges” I refer to the spaces or domains within which each vertex is allowed to move while avoiding vertex eversions (which would describe unwanted inverse spaces). This eversion problem would, for instance, occur if vertices \{1\} and \{9\} in the middle of figure 3.6 were to change their positions in such a way that \{9\} was located above \{1\}; the cellular space between both vertices would evert and result in an undefined space\(^8\).

I saw the resulting flexibility of cellular shapes in analogy to the irregular geometric assemblies observed in natural cellular tissues. However, not all parts of the grid topology allow for parametric manipulation equally. Six of the fourteen vertices of each dodecahedron are parametrically over-constrained and require a geometrically awkward workaround. As mentioned before, the programming of

\(^8\)The vertex numbers 0-13 used in Zellkalkül have in later developments been adjusted to the numbers 1-14 as shown in section E.1 in order to improve compatibility with computational geometry software such as Surface Evolver and Qhull.
3. TOOLMAKING STUDY

Figure 3.6: Rhombo-dodecaherdal cellular geometry (left), face triangulation and vertex ranges (centre), STL format illustration (right).

Figure 3.7: Tetrahedral and octahedral movement ranges of two vertex types.

Figure 3.8: Detail of parametrically altered close-packed cells. Digital rendering (left) and rapid prototype wax model (right).
intended developmental processes from the “inside-out” viewpoints of a digital zygote and its descendants is a challenging task for the user of the tool. This task was now further complicated by requiring yet more differentiated control to be exerted from this difficult perspective.

With respect to the secondary research question of which geometric principles allowing parametric shape control could be incorporated into packings of rhombo-dodecahedral cells, I implemented and demonstrated one of a number of potential candidates. Some geometric elements are over-constrained in this solution, somewhat limiting parametric flexibility. These results do not seem to offer a practical solution to the problem of jagged surfaces and further complicate scripting efforts. Results from this stage of the toolmaking study raise the question of how geometric principles and strategies originate. This question is further addressed in the reflective analysis in chapter 4.

3.6 Re-orientation towards foam-like grids

The secondary research question investigated at this stage of the toolmaking study is: Can a visually irregular and rationalised spatial tessellation be developed based on the previously explored packing of rhombo-dodecahedral cells?

Research objectives at this stage did not have an immediate impact on the primary research questions, but they build upon the investigation of the previous toolmaking stages and further move the toolmaking study towards a project-driven approach.

Based on the toolmaking work undertaken so far, I came to the conclusion that the conventional pattern of making generic tools for problems with known solution strategies is not likely to contribute to or to inform toolmaking for designing. It became increasingly evident to me that the goals for design toolmaking should
be explored in applied design contexts rather than fixed \textit{a priori} and removed from applied designing. In order to contribute practical value to designing, the operations supported by a design tool should address design challenges as they are posed by conflicting design requirements in applied design processes. I thus sought involvement in applied design projects.

The first such project offering itself was the space grid design for the Beijing National Swimming Centre (by PTW, ARUP Sydney, CSCEC; see figure 3.9). When I became involved with the geometry rationalisation problem posed by the project, both stages of the project competition had been judged and the decision for a large, box-shaped building resembling a “fish tank filled with irregular liquid foam” had been made. The design team had so far illustrated this concept using two-dimensional images of foam which were used to visualise an irregular space frame structure supporting the Beijing National Swimming Centre as well as creating a layered, climate-controlling enclosure.

![Figure 3.9: Beijing National Swimming Centre (PTW, ARUP, CSCEC) as displayed in the winning entry to the first competition stage.](image)

A suitable grid structure had to be identified to guide the development of a structural solution for the building enclosure. Apart from functioning structurally, it was required to satisfy the contradicting criteria of cost-efficient regularity in manufacturing and construction on the one hand and visual irregularity resembling liquid foam contained in a box shape on the other. Posing a challenge with no known solution strategy, the project presented a new challenge to relate to, which
remained important during all following stages of the toolmaking study. The grid structures that I sought from this point onwards needed to be visually irregular (i.e. appear non-repetitive) while at the same time composed from a set of components comprised of as few as possible edge lengths and vertex angles so as to harness economic benefits in the batch production of strut and node elements for respective space frame structures (that is, to draw economic advantages from repetition). My attention now moved away from software development technicalities towards finding possible geometric strategies and strategies of integrating potential tools into the design process. These issues now became the focus of my work, and software development became a means to an (unknown) end. From this point onwards, I engaged in tool development only as a result of identifying a potential strategy first. This contrasts with earlier stages of the toolmaking study in which I developed tools without applied projects to relate to.

Looking back at the software developed after this point of the toolmaking study, a change in the quality of my programming work becomes visible. While Zellkalkül was written in Java™, a relatively low-level programming language, with considerable attention to stability, reliability and a robust user interface, most of what I programmed after this point was put together in a “quick and dirty” fashion in higher-level scripting languages. I now paid less attention to those aspects of software that allow use by others and concentrated on fast evaluation of ideas and quick results. The previous programming work could probably best be compared to the painting of a large, carefully composed picture while my programming after this point became sketchy⁹, probably with little meaning or immediate value to others. In short, the previous priority on tools over outcomes in my work was from now on reversed.

⁹One exception is the “soap bubble tool” discussed in section 3.11.
3. TOOLMAKING STUDY

As discussed in section 1.4.3, the reconciliation of conflicting geometrical design requirements for achieving buildable solutions is referred to as geometry rationalisation. This work was taken care of by the ARUP engineers who were part of the Swimming Centre’s design team. The rationalisation approach taken by ARUP was based on a foam structure previously presented by Weaire and Phelan [1994] in response to the so-called Kelvin Problem (see section 1.4.1). This approach initially promised some degree of visual irregularity based on very few different strut and node types (see Bosse [2004]). However, the expected efficiency of this rationalisation approach had to be largely renounced later when the Weaire-Phelan-like structure was rotated to achieve greater visual irregularity and then cropped to form planar exterior and interior surfaces as illustrated in figure 1.21.

Searching for alternative solutions in response to the Beijing project, I studied the rhombo-dodecahedral grid used in Zellkalkül as a potential solution for this project. The result is an irregular octet-truss derivation reported in Fischer [2005], which does, however, not circumscribe convex polyhedra and thus does not resemble liquid foam as was required for the structure of the Beijing National Swimming Centre. As this approach forms the basis for my work presented in the following section 3.7, I am including a brief description of this structure here.

The development of this irregular space grid with a limited number of components starts from the rhombo-dodecahedral cell shapes used in Zellkalkül. Figure 3.10 shows the relationship between close-packed spheres and rhombic dodecahedra (see Frazer [1995] and Fischer et al. [2005]).

After the grid of rhombic dodecahedra is derived from packed spheres, two operations are subsequently applied. First, the grid is modified to appear visually more irregular while minimising the number of strut lengths and node types by moving centre points away from their original location to reduce the amount of
symmetry within each cell. Second, it is divided into identical partitions, which can be broken down into self-similar sub-partitions. I refer to these units as “rhombic pyramids”.

Figure 3.12 shows how the rhombic dodecahedron can be broken into twelve sub-cells. The twelve faces of the dodecahedron are the bottom faces of twelve rhombic pyramids whose top vertices meet at the centre point of the dodecahedron.

Figure 3.11: Wooden model showing break-down of rhombic pyramids into self-similar units.

A rhombic pyramid can be broken down into self-similar cells at a volume ratio of 1:8. Sub-units are combined at different rotation angles. This results in a multiplication of strut angles within the structure and introduces greater visual irregularity when rhombic pyramids assemble into larger (or break down into smaller) composite units (see figure 3.12).

In order to make this system useful for achieving visually irregular grid structures, it might be desirable to allow for different kinds of variation and change within this structure. One possible starting point (which I do not pursue in detail
3. TOOLMAKING STUDY

**Figure 3.12:** Decomposition of dodecahedra into self-similar rhombic pyramids.

here) is offered by the observation that, after one level of fractal decomposition, two adjacent pyramids of two dodecahedra produce a new rotated rhombic dodecahedron at a smaller scale (see left of figure 3.13). The remaining three illustrations in figure 3.13 show how fractal resolution to different levels of depth offers design choices and how this resolution can be spatially modulated to produce varying densities, for example for visual or structural purposes.

**Figure 3.13:** Decomposition and variation in (local) density.

A comparable effect in two dimensions can be observed in the fractal decomposition of the pinwheel pattern found on the façade of *Federation Square* (Lab architecture studio) in Melbourne (see figure 1.16).

Figure 3.14 shows that the dodecahedron’s centre point, to which the decomposition process refers, can be moved in order to achieve greater visual variety. However, this increases the number of required beam lengths and node angles considerably. In order to achieve the increase in visual irregularity obtained by moving the centre point while minimising the increase in beam lengths and node
angles, I decided to search for points at which the negative effect on the rationality of this grid might be minimal. The outer-left diagram in figure 3.14 shows the rhombic dodecahedron, made up of twelve rhombic pyramids meeting at the original centre point. The second-left diagram shows the search space within which I scanned for alternative points at which the increase in number of different required strut lengths is minimal\(^\text{10}\).

**Figure 3.14:** Search space for alternative midpoints and dodecahedral grid with shifted midpoint.

The above-described scan revealed the existence of minima in the number of resulting beam lengths within the marked search space. Table 3.1 shows the most successful results obtained by scanning the space in all three dimensions using three nested loops at a step width (or resolution) of 0.001, assuming that the dodecahedron corresponds to a unit sphere with diameter 1.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>strutlengths</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.198</td>
<td>0.28</td>
<td>0.198</td>
<td>22</td>
</tr>
<tr>
<td>0.197</td>
<td>0.293</td>
<td>0.197</td>
<td>23</td>
</tr>
<tr>
<td>0.198</td>
<td>0.16</td>
<td>0.198</td>
<td>23</td>
</tr>
<tr>
<td>0.198</td>
<td>0.164</td>
<td>0.198</td>
<td>23</td>
</tr>
<tr>
<td>0.198</td>
<td>0.26</td>
<td>0.198</td>
<td>23</td>
</tr>
<tr>
<td>0.199</td>
<td>0.151</td>
<td>0.199</td>
<td>23</td>
</tr>
<tr>
<td>0.2</td>
<td>0.164</td>
<td>0.2</td>
<td>23</td>
</tr>
<tr>
<td>0.205</td>
<td>0.284</td>
<td>0.205</td>
<td>23</td>
</tr>
<tr>
<td>0.206</td>
<td>0.191</td>
<td>0.206</td>
<td>23</td>
</tr>
</tbody>
</table>

\(^{10}\)I did not evaluate the number of different node types. A discussion of my plans to develop a node-counting program is given is section 3.9.
The most successful result was found at point \( \{0.198, 0.28, 0.198\} \) as illustrated in the two right diagrams of figure 3.14. Figure 3.15 shows the structure resolved to two levels with moved midpoint (left), with variable density (centre) and with the original rhombo-dodecahedral frame removed. As with the original dodecahedra, these structures can be combined to fill space in a visually irregular fashion. The dodecahedral unit with off-centre mid-point loses some of its previous symmetry and allows for combinations in different orientations within close packing, however at the cost of an increasing number of required node types.

![Figure 3.15: Moved midpoint, variable density, removal of dodecahedral frame.](image)

The results I obtained scanning for alternative midpoints depend greatly on the degree of mathematical precision applied when counting the number of different strut lengths contained in the structure. Generous rounding naturally produces smaller numbers of resulting beam lengths and construction systems that are
designed to bridge tolerances can compensate for this generosity. Truss systems in which beams are connected to nodes, for example by driving threaded bolts out of the beam end into the nodes (see figure 3.33), can offer an elegant means to compensate for low generative precision. Another practical response to this precision issue is to set the step width to a small distance that can expected to offer “good enough” (or “precise enough”) results for a given scale and given tolerance requirements. Talking to experts in the field\textsuperscript{11}, I learned that $\frac{1}{1000}$ of a millimeter should be a sufficiently precise tolerance level. Hence, pushing precision issues below this tolerance level can be expected to offer “good enough” results in this sense. However, investigating space grids at a basic level without a concrete project to relate to, models of grid topologies do not (yet) have a scale, unless an arbitrary one is chosen. Thus, the above tolerance level has no reference point to relate to.

A more difficult problem seems to be caused by the step width at which the search space is scanned. No matter how small the step width is chosen, there will always be an infinite number of untested points between any two tested points, at which more successful results could potentially be located. As the sampling does not proceed in a continuous way, interesting points could be missed. A response to the problem caused came to mind, which I did not find the time to test

\textsuperscript{11}While presenting intermediate results of this study to senior engineers at MEROTSK in Würzburg, Germany, I asked to what tolerances space frame structures are manufactured and constructed. The immediate answer was: “There is no tolerance! Everything is made precisely and it just fits.” After repeated asking and some discussion amongst the engineers it was then established that the company uses a custom-developed planning software which, as the manufacturing facilities, works to a tolerance of $\frac{1}{1000}$ of a millimeter, taking into consideration temperature differences between prefabrication and construction sites and resulting thermal expansion.
after the project discussed in section 3.7 emerged. In this approach, points in the search space would be evaluated, and using a three-dimensional modelling tool, visualised in a colour corresponding to the result of the evaluation. This might offer a data visualisation in which more interesting areas and neighbourhoods of minima could be recognised (provided results are distributed in sufficiently ordered gradients).

The above-described grid system I developed during this stage of the toolmaking study represents a positive answer to the secondary research question investigated at this stage, which asked whether a visually irregular spatial tessellation be derived from the previously explored packing of rhombo-dodecahedral cells. This system does however not resemble liquid foam. Research objectives at this stage did not have an immediate impact on the primary research questions, but, similar to the previous stage, they build upon the investigation of the previous toolmaking stages and move the toolmaking study towards a project-driven approach.

3.7 Self-supporting façade system

The secondary research question investigated at this stage of the toolmaking study is: Can the geometry rationalisation approach developed during the previous stage of the toolmaking study (see section 3.6 above) be modified to form a structural façade system that casts tree-like shadows?

Similar to the previous stage, research objectives at this stage did not have an immediate impact on the primary research questions, but they build upon the investigation of the previous toolmaking stages and further move the toolmaking study towards a project-driven approach.

As mentioned in section 3.6, the above-described visually irregular space grid system does not circumscribe convex polyhedra and thus does not resemble liquid
foam as was required for the structure of the Beijing National Swimming Centre. However, I used the basic topology of the system as a starting point for another applied project, Metropol Parasol (Jürgen Mayer) in Seville, Spain, at a stage when a semi-permeable, self-supporting free-form façade for a conglomerate of several large, mushroom-like pavilion structures was under consideration. Apart from acting as a visually irregular structure, the façade should cast an irregular shadow, similar to the shadows cast by trees.

In my proposal, the Metropol Parasol shapes are approximated by close-packed spheres, which are translated into the previously explored packed rhombic dodecahedra. These in turn are broken down into irregular tetrahedra. Based on my previously developed code and geometry I was able within a short time frame to resolve the packed dodecahedra into rhombic pyramid units made of folded sheet metal (see centre and right of figure 3.16), which can be assembled into semi-permeable structures of irregular appearance. Each tetrahedron is circumscribed by a piece of sheet metal, which is bent to form two or three triangular faces of the tetrahedron. Each triangular face contains a cut-out shaped to maximise openness to allow light to penetrate partially (see figure 3.16). By resolving the overall Parasol structures at different scales, different densities can be generated to achieve the desired shadow-casting effect.

Figure 3.16: Spheres packed to form Metropol Parasol shapes, irregular tetrahedra made of sheet metal and node connection detail.
According to the rationalisation approach presented in section 3.6 above, the visual irregularity of this structure can be increased by moving the centre points of rhombic dodecahedra. From this, the same above-mentioned rationalisation effect can be expected to allow for economic batch-production of components. Also similar to the approach presented in section 3.6, it is possible to replace each tetrahedron with eight tetrahedra of a smaller scale (see figure 3.11). These in turn can be replaced in the same manner and so forth so as to modulate visual irregularity, local density and possibly structural properties.

As shown in the above Metropol Parasol scheme on the left in figure 3.16, and as noted in sections 3.2 through 3.5, the exterior surface of a shape approximated by packed rhombic dodecahedra is “jagged”. The resolution into tetrahedra of different scales allows for the reduction of this effect and the closer approximation of the intended shapes. It is for example possible to fill up “jagged” surfaces using tetrahedra and “half dodecahedra” (which can be composed from irregular tetrahedra) as shown in image 3.17. This principle can also be utilised to construct walls with a thickness of half of the diameter of the underlying unit sphere, thus reducing the façade thickness. To allow varying numbers of folded metal sheets to meet at node points at different angles, I proposed the use of identical spherical steel units as a single, universal node type (see right of figure 3.16).
My software development work at this stage of the toolmaking study followed preceding strategic ideas and served purposes of evaluation, productivity and illustration to communicate possible outcomes. In contrast to previous work undertaken in this study from a toolmaker’s perspective, I now assumed the designer’s perspective. I did not aim to support anybody else designing but designed myself, developing a proposal for a structural solution in parallel with the project engineers who, after testing tens of their own ideas, received my proposals positively (see email by Tristram Carfrae in appendix J). I did not develop a piece of software that could be used as a tool to solve similar problems in other contexts.

Over the course of this stage of the toolmaking study, I developed a proposal for a potential solution to the initial secondary research question of whether the geometry rationalisation approach developed during the previous stage of the toolmaking study (see section 3.6) could be modified to form a structural façade systems that casts a tree-like shadow. This proposal was assessed positively by engineers working in the context in which the problem originated, my work leading towards it however can not be described as toolmaking. At this stage of the toolmaking study, my focus was on providing specific design proposals. I designed a structural geometry, not a tool that could be used to design structural geometries.
3.8 Development of irregular space grids

The secondary research question investigated at this stage of the toolmaking study is: Can the approaches described as pre-, post and co-rationalisation offer design procedures through which rationalised space grid structures resembling liquid foam could be achieved?

The primary research questions addressed by observations made during this stage of the toolmaking study are question 2: Can the development of digital design tools be justified after the abandonment of design methods? and question 4: Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?

Searching for more appropriate solutions to the Beijing Water Cube problem that could accommodate both irregular appearance and a rationalised structure to allow cost-effective manufacturing and construction while resembling liquid foam, I pursued toolmaking with less priority. A large part of the investigation now focused on hands-on geometric enquiries based on methods such as sketching or just thinking. Another approach was to use existing digital tools made by others, mainly for computing convex hulls and minimal surfaces. This led to some new ideas based on systematic combination of convex polyhedra and on “evolving” three-dimensional Voronoi diagrams to approximate liquid foam geometry.\(^{12}\)

Figure 3.19 shows a three-dimensional Voronoi diagram inscribed into a cube shape. It was generated from a set of 150 seed points that were first placed inside a 14cm x 14cm x 14cm volume at a density of no less than 5mm distance between any two points. Then, a Voronoi diagram was generated from the set of seed points using the program Qvoronoi of the Qhull software package. Peripheral cells in a

\(^{12}\)This approach is also reported in sections 2.1 and 2.2 of Fischer [2007b].
Voronoi diagram are open or “unbounded” with “missing” vertices said to lie “at infinity” or “at an infinite distance”. To achieve flat surfaces around the Voronoi-diagram, all vertices within 3cm from any point of the surface of the 14cm x 14cm x 14cm cube were “pulled” onto the surface of a smaller cube volume of 8cm x 8cm x 8cm. The resulting space grid was then processed using the software *Surface Evolver* (see Brakke [1992]).

*Surface Evolver* allows the simulation of the surface-tension minimising behaviour of dry liquid foam. My aim was to minimise differences between the angles at which edges in the Voronoi structure meet. I hoped this could reduce the number of contained vertex angles and increase the occurrence of node conditions that are approximated by dry liquid foam, the so-called caltrop configuration of 4 edges meeting at angles of 109.47°. As illustrated in figure 3.20, this procedure did not produce the desired result.

Differences between vertex angles did indeed “evolve” to a decreased level, but only to a point beyond which liquid foam would change its topology for instance by changing relative positions of bubbles, by popping or by merging bubbles. Topological changes of this kind are however not (currently) supported by Surface Evolver. Even if this behaviour was supported, it would still offer only limited
utility in the grid rationalisation aimed for here. It would only minimise the number of angles meeting at shared vertices (i.e. node types in corresponding space frames). It appears highly unlikely that a three-dimensional grid structure with only $109.47^\circ$ angles only could be found (unless struts are curved – see figure 3.26). Controlling the number of different edge lengths would remain a difficult task. An additional challenge to this approach is posed by a limitation of Voronoi diagrams. These can only be generated for convex shapes. The procedures known to produce Voronoi diagrams$^{13}$ fill hollow spaces defined by concave surfaces (such as those forming interior architectural spaces) with cells. This requires a separate treatment of individual, non-concave shapes such as individual walls and floors, as well as some way to co-ordinate vertex points where those elements are to be re-joined.

While the above-described approach aims at generating an irregular grid topology first and then modifying it so as to conform to (rational) configurational requirements, the following approach involves combination of convex polyhedra and considers configurational laws before a grid topology is assembled. Based on the observation that with $120^\circ$ angles and one strut length only, it is possible

\[\begin{align*}
\text{Figure 3.20: Processing of Voronoi foam in } & \text{Surface Evolver.}
\end{align*}\]

\[\begin{align*}
\text{3. TOOLMAKING STUDY}
\end{align*}\]

\[\begin{align*}
\text{utility in the grid rationalisation aimed for here. It would only minimise the number of angles meeting at shared vertices (i.e. node types in corresponding space frames). It appears highly unlikely that a three-dimensional grid structure with only } 109.47^\circ \text{ angles only could be found (unless struts are curved – see figure 3.26). Controlling the number of different edge lengths would remain a difficult task. An additional challenge to this approach is posed by a limitation of Voronoi diagrams. These can only be generated for convex shapes. The procedures known to produce Voronoi diagrams}\text{ fill hollow spaces defined by concave surfaces (such as those forming interior architectural spaces) with cells. This requires a separate treatment of individual, non-concave shapes such as individual walls and floors, as well as some way to co-ordinate vertex points where those elements are to be re-joined.}

While the above-described approach aims at generating an irregular grid topology first and then modifying it so as to conform to (rational) configurational requirements, the following approach involves combination of convex polyhedra and considers configurational laws before a grid topology is assembled. Based on the observation that with } 120^\circ \text{ angles and one strut length only, it is possible}

\[\begin{align*}
\text{13 There are different known strategies that can be used to translate sets of seed points into Voronoi diagrams, the so-called “plane sweep” method, “divide and conquer” method and “Fortune’s algorithm”.}
\end{align*}\]
to compose grids containing convex polygons of different sizes (see left of figure 3.21), I tried to achieve a structure with similar characteristics in three dimensions.

My idea was to start from a large convex polyhedron which contains only one node angle and one strut length and to group smaller polyhedra around it, hoping to achieve a three-dimensional space-filling system analogous to the two-dimensional one shown on the left of figure 3.21. As the large convex polyhedron, I chose a truncated icosahedron (the geometry of the C$_{60}$ molecule discussed in section 1.5.3). Trying to fit smaller polyhedra onto its surface and filling as much space as possible, the truncated octahedron (a Kelvin cell with straight edges) offered the result shown on the right of figure 3.21. Multiples of the shown clusters can be assembled in a three-dimensional Cartesian fashion, however leaving some unfilled space between them$^{14}$. As shown on the left of figure 3.22, the faces of the smaller polygons do not line up entirely, leaving a small opening. By introducing a second strut length and a second node type, this gap can be closed (see right of figure 3.22). The slightly increased number of different node and strut types is still very low and probably suitable for batch production if considered for an applied space frame design project.

---

$^{14}$I later found a similar, yet different grid structure (based on other polyhedra) designed as an exercise at the Hochschule für Gestaltung in Ulm in the late 1950s (see Lindinger [1991], p. 138)
At this point I had designed two grid systems (shown in figures 3.19 and 3.21), both of which had specific shortcomings. Nevertheless, the procedures underlying the two systems can be described as post-rationalisation in the case of the evolved Voronoi grid (see figure 3.19) and pre-rationalisation in the case of the assembled polyhedra (see figure 3.21).

This reinforced my interest in co-rationalisation, as introduced to me by Robert Aish (see section 1.4.4), which was also related to the notion of embedded rationale by Hugh Whitehead in an email exchange initiated by myself when considering the notion of co-rationalisation (see email by Hugh Whitehead in appendix K).\textsuperscript{15} According to Whitehead, an embedded rationale offers means for variation and exploration (for example by parametric means) while possible outcomes conform to initially known material and construction constraints. Co-rationalisation is described as a process in which the compositional system is defined alongside and through the process of designing a form. This seems to indicate a new quality in design rationalisation (see section 1.4.3)\textsuperscript{16}.

I soon realised, however, that this association of previous work with the two

\textsuperscript{15}See email by Hugh Whitehead in appendix K
\textsuperscript{16}I discussed the design processes underlying these structures as having followed pre- and post-rationalisation approaches in Fischer [2005]. In this paper I used the terms “bottom-up” and “top-down” for what I am referring to here as pre-rationalised and post-rationalised.
kinds of rationalisation in itself can be described as post-rationalisation. I began wondering if this distinction could indeed be of prescriptive value at the outset of a project as terms like “strategy”, “approach” or “method” suggest. Can this distinction inform designers in what they do before there are any strategic ideas? Is any one of the three kinds of rationalisation more worthwhile under different circumstances than the others? Before the application of a new idea regarding the procedure of designerly action, can it be known to which of the three approaches the procedure will eventually turn out to correspond?

This experience was reminiscent of my experience of trying to apply Zellkalkül (see section 3.2 through 3.5). Back then, I had knowledge of biological models of cellular development. These models were useful in explaining and in understanding observed biological phenomena but they turned out to be insufficient in my attempt to automate cellular development digitally. Apparently, the model had descriptive capabilities but it seem to fail when being applied prescriptively. Now, again, I had a model (the distinction between different modes of rationalisation), which seemed useful in offering a post mortem descriptions of my grid rationalisation work. But when applied to prescribe parts of design processes in advance, this model seemed to fail, too.

In parallel to my thinking about rationalisation strategies, I also considered possible ways of assessing visual irregularity. One thought was to develop a way of assessing visual irregularity of space grids resembling liquid foam in a quantitative analysis procedure, which could potentially be automated. A procedure of this kind would start from circumferential space grids (see right of figure 0.1), which contain convex, soap-bubble-like polyhedral cells. Then, I thought, the variation of bubble sizes within the structure could be assessed. This could be achieved by determining the centre point of each cell as well as the centre point of each
cellular face that is defined by more than three edges. With these points, it would be possible to decompose every cell into tetrahedra, whose volume could then be determined in a straightforward fashion. By adding up all partial tetrahedral volumes of every cell, the total volume of every cell in the respective grid structure could be determined and then the variety (or standard deviation) amongst the volumes of all cells could be calculated. The underlying assumption was that less variety amongst cell sizes would indicate less visual irregularity and vice versa. Once the variety of cell sizes was determined, it might be possible to relate this measure to the number of different strut lengths and node types contained within a respective structure, which could be used to assess the degree of rationalisation of visually irregular, foam-like space grids for economic batch production.

I dismissed this plan soon after modelling some known, foam-like grid structures such as packed truncated octahedra (see section E.2 in appendix E) and Weaire-Phelan-like grids (see sections E.3 and E.4 in appendix E) as possible test cases. Packed truncated octahedra contain only one node type, only one strut length and a single cell volume. Weaire-Phelan-like packings contain multiple strut lengths and multiple node types and cells of two different volumes\textsuperscript{17} and hence are said to appear more irregular than truncated octahedra\textsuperscript{18}. Visual inspection of three-dimensional grid models, however, shows that their visual irregularity depends to a significant degree on the viewer’s perspective. The left and the centre of figure 3.23 both show the same grid of packed truncated octahedra. On the left, the grid appears more ordered while at the centre it appears visually more irregular and

\textsuperscript{17}Weaire-Phelan-like packings contain cells with two different volumes before being “evolved” into Weaire-Phelan foam with curved edges, see section 1.4.1

\textsuperscript{18}This is the reason why the Weaire-Phelan like grid was preferred over packed truncated octahedra in the geometry rationalisation for the Beijing National Swimming Centre (see section 1.4.3).
somewhat more like liquid foam. On the right of figure 3.23 is a Weaire-Phelan-like grid with the same number of cells. Even though this structure is described as visually more irregular than packed truncated octahedra and even though it contains greater sets of different edge lengths and node types, it appears, from this perspective, visually less irregular than the truncated octahedra shown at the centre. The depth of grid structures has a similar effect as perspective. It can greatly increase visual irregularity, also resulting in truncated octahedra appearing more irregular than Weaire-Phelan foam.

Figure 3.23: Visual irregularity in different space grids and from varying perspectives.

Counter-intuitive effects of this kind demonstrate that the above-described procedure for quantitative assessment of visual irregularity and geometry rationalisation cannot be reliable as it ignores important aspects concerning the observer’s viewpoint and grid depth (as well as changes in grid depth). For this reason, I have abandoned the idea of assessing visual irregularity and geometry rationalisation on a purely quantitative basis.

At an early stage of the toolmaking study, I was surprised to read that irregular liquid foam tends to form unified “tetrahedral” vertex angles of 109.47° (see section 1.4.1 and Weaire [1999], pp. 24-26). Finding this phenomenon difficult to imagine, I decided to explore it physically. For this purpose, I made a modelling kit from bamboo kebab sticks of irregular lengths and wooden spheres (Chinese moth balls), with four drillings at tetrahedral (109.47°) angles. From this kit, I made the model shown in figure 3.24, which shows that irregular grids resembling liquid foam with
curved edges can be formed with tetrahedral node angles only.

**Figure 3.24:** Stick model of irregular liquid foam made of kebab sticks and wooden moth balls.

Now, developing irregular space grids, I began considering the use of Kelvin-like cells (truncated octahedra, see section E.2 in appendix E) and Weaire-Phelan like cells (see sections E.3 and E.4 in appendix E). To develop an understanding of these structures and their spatial properties, I made a physical model of them (see figure 3.25), beginning with stick models of packed truncated octahedra. For these stick-models I re-used the wooden nodes with the tetrahedral node angles of the earlier stick model, and combined them with sticks of a unified strut length as is found in grids of packed truncated octahedra.

**Figure 3.25:** Packing of truncated octahedra (left) and of Weaire-Phelan-like cells (right).
This packing structure (see left of figure 3.26) contains square polyhedron faces, resulting in one node angle of $90^\circ$ at each node. I knew that the wooden node elements would not fit into this structure without applying some force and bending the struts (see centre and right of figure 3.26). I was aware that with only tetrahedral node angles I could achieve a structure that was topologically equivalent to a grid of packed truncated octahedra and sufficiently similar to it to learn about this structure. I thought it might be interesting to see what would happen if the structure were forced to contain $109.47^\circ$ angles only. The result is a curvature in each strut that is not unlike the Plateau border curvature observed in natural liquid foam (see left and centre of figure 3.26 and section 1.4.1). The bending of the elastic bamboo sticks results in a uniform pre-stressed configuration around each node, seemingly resulting in interesting overall structural properties. Each truncated octahedron seems to form a closed, pre-stressed structural element. Once combined, multiple such elements interact and display a remarkable physical stiffness. It can be assumed that by using the strut lengths found in Weaire-Phelan-like foam (see sections E.3 and E.4 in appendix E) a similar edge curvature and similar structural effects could be achieved on the basis of a Weaire-Phelan-like grid topology.

Having physically built and experienced this structure, I wondered if the structural properties of these pre-stressed space grids could be compared to say conventional octet truss grids by means of computer-aided structural analysis. The non-standard nature of the new pre-stressed space grids, however, would make modelling for computer-aided structural analysis purposes challenging. In general, the outcomes of structural simulation are models of stressed and deformed structures. The structures in question, however, are stressed by deformation already in the initial state. In this way, they start in the kind of configuration
that is usually the outcome of structural analysis. Modelling the new space grids for this purpose would likely require some simplification, by which pre-stressed and elastically deformed elements are replaced with non-stressed, permanently deformed elements, and assigned bending forces similar to those of the stressed and deformed ones. Those in turn could only be known after physical measurement. Following a simulation of a simplified model, it would then be necessary to validate the obtained structural simulation results by testing analogous physical models – similarly to the physical prototyping and testing of non-standard building elements, which is a common procedure in non-standard building practice). Physical measuring and testing would thus have to precede and to follow a reliable computer-aided structural simulation.

![My bedspring: patch, stressed node, unstressed node.](image)

**Figure 3.26:** My bedspring: patch, stressed node, unstressed node.

With respect to the secondary research question of whether the approaches described as pre- post and co-rationalisation offer design procedures through which rationalised space grid structures resembling liquid foam could be achieved, observations made during this stage of the toolmaking study suggest a mixed answer. It is possible to categorise the undertaken geometry rationalisation work as pre- or post-rationalisation. In some cases, however, an exclusive categorisation into just one of the three approaches seems difficult. More importantly, any categorisation seems possible only in retrospect. This began raising the question
of whether the different rationalisation approaches could offer guiding principles by which designers could proceed in rationalising architectural geometry.

3.9 Exploring co-rationalisation

The secondary research question investigated at this stage of the toolmaking study is: Can the integration of non-design tools as “building blocks” into design tools constitute a viable approach to toolmaking for design?

This stage of the toolmaking study directly addressed primary research question 4: Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?

Co-rationalisation or embedded rationale, as introduced in the previous section, had so far not received much research attention and I had not yet attempted to co-rationalise foam-like grid structures. As discussed in section 1.4.4, the example of Gaudi’s hanging model incorporated a self-organising capability to constrain form exploration to structurally desirable solutions, which could be referred to as an embedded rationale and could thus be related to the concept of co-rationalisation. I decided to develop a similar strategy for the rationalisation of foam-like grids in software in preparation for a possible toolmaking effort that could support this approach. In analogy to the strings in Gaudi’s hanging model and to the torus patches into which the roof surface of the Sage Music Centre (Foster and Partners) was resolved (see appendix K), I reasoned that a co-rationalising capability should be distributed throughout the geometrical system in question and embedded into its constituent elements. In the case of foam-like space frames these elements are obviously the packed bubbles. Every bubble should thus have an embedded “co-rationale” to enable it to negotiate its relationship with each neighbour to achieve a rationalising effect.
3. TOOLMAKING STUDY

My first attempt to achieve such a space grid co-rationalisation approach takes as a starting point the tendency of Plateau borders in three-dimensional liquid soap to meet at tetrahedral angles of 109.47°. From this tendency I inferred that any two adjacent soap bubbles should connect in such a way that the distance between their centre points (see points A and B in the diagrams shown in figure 3.28) is given by the term

\[ d_{AB} = \sqrt{(r_2 - \frac{r_1}{2})^2 + \frac{3}{4} \cdot r_1^2} \]

where \( r_1 \) and \( r_2 \) are the radii of the two bubbles\(^\text{19}\). I thought that co-locating any two adjacent bubbles in a foam at the distance given by this term would maximise the occurrence of 109.47° angles. I also assumed that if the bubbles contained within and added to such a structure were to have a very limited set of diameters, there would be only a very limited set of combinations of adjacent bubble diameters. Once such an assembly of bubbles were translated into a circumferential space grid, there would be a limited set of resulting strut lengths and node types, possibly allowing co-rationalisation for the economic batch production of building elements for space frames derived from the grid.

Figure 3.27 shows a test foam of the described kind. It was generated by a “swarm” of mobile bubbles, which were placed at random centre points within an overall box shape. The bubbles have the capability to seek neighbouring distances according to the above-mentioned formula and were initially placed randomly in a box-shaped space and subsequently moved individually and iteratively to comply with the distance rule expressed in the above term with their adjacent neighbours.

Figure 3.27 shows that neighbouring bubbles approximate the desired distance. Many neighbouring bubbles meet in such a way that their surface tangents are

\(^{19}\)This formula was developed with valuable support from Cristiano Ceccato.
close to $120^\circ$, quite similar to clusters of soap bubbles. The rationalising effect of this approach is nevertheless rather moderate. Distances between bubbles change towards the desired condition, but reach it in only few cases. I set up the movement procedure in such a manner that once the desired bubble distance has been reached (including just partial fulfillment and with changing levels of tolerance), the procedure would terminate. The movement process, however, never reached this stage.

The following is a brief discussion of the explanation I found for the lack of success of this rationalisation approach, according to which it is difficult to achieve the conditions described by the above-discussed distance formula everywhere in larger clusters bubbles. In order to derive a circumferential grid from a cluster of bubbles, it would be necessary to identify the vertices between all tetrahedral neighbourhoods of any four bubbles. In two dimensions, this vertex context can be illustrated with three circles in a triangular configuration as shown in figure 3.28.

![Figure 3.27: Bubble distances evolved to comply with distance rule.](image)

**Figure 3.27:** Bubble distances evolved to comply with distance rule.

![Figure 3.28: Interdependence of vertex locations between multiple bubbles aiming to achieve angles of $109.47^\circ$.](image)

**Figure 3.28:** Interdependence of vertex locations between multiple bubbles aiming to achieve angles of $109.47^\circ$. 

171
3. TOOLMAKING STUDY

The diagram on the left of figure 3.28 shows two bubbles at the described distance projected onto a two-dimensional plane with their centre points A and B located at the distance given by the above term. The bubble surfaces (in the two-dimensional diagram represented by circles) meet at 120°, which in two dimensions, corresponds to the desired 109.47° angles in three dimensions. The vertical line between the two circle intersection points corresponds to the circular soap film found between two touching three-dimensional bubbles. The diagram at the centre of figure 3.28 shows how the situation changes when more than two bubbles meet. A third bubble around centre point C is added and placed to conform with the distance rule with respect to both other centre points A and B. The two intersection points between any two of the three bubbles can be connected to form straight lines, which again correspond to the circular soap films between two analogous bubbles in three dimensions. This appears useful for identifying the vertex points between the spheres (or circles) as all three straight intersection lines in the centre diagram of figure 3.28 meet in one single point. Analogously, the four circular intersection areas found in a three-dimensional cluster of four spheres also intersect in one single point. This point, however, turns out to coincide with the required vertex point only if all four bubbles (three spheres) have the same size. In a foam of different bubble diameters the strut angles around this vertex will not be equal and struts will hence not meet at this vertex. The location where all strut angles are equally 109.47° (120° in two dimensions) is shown in the diagram on the right of figure 3.28. This point was found through iterative approximation of the smallest total strut length between points i, j and k, as it is determined by natural foam’s tendency to minimise its surfaces. This vertex point depends on the location of the remaining outer bubble intersection points i, j and k. The addition of new bubbles near points i, j and k will move the respective intersection point and turn it
into a new vertex. The location of the vertex between the three bubbles around A, B and C will be influenced. Ultimately, the positions of all vertices in a foam of this kind are therefore interdependent. It seems impossible to change the conditions of one vertex or edge without changing those of the others. In a self-referential way, changing the conditions of one element changes other elements, on whose conditions it depends itself. I therefore decided to look for an alternative strategy.

Based on Whitehead’s description (see appendix K) of how an embedded rationale was utilised in the rationalisation of the façade of the Sage Music Centre, I focused more on thinking about fixed grid topologies. These should allow for the addition or subtraction of individual bubbles and support some form of parametric variation that was constrained to a limited set of desired points that are known to offer rationalisation benefits. If cellular relationships within foam-like grids could be constrained to a limited set of conditions that was known to be advantageous, then the exploration of variations based on such limited sets could allow co-rationalisation in analogy to Gaudí’s hanging model. Just as the hanging model constrains vault curvatures to catenary curves (see figure 1.17), this approach should constrain possible variety in bubble configurations to those that result in small numbers of different edge lengths and node angles. In this approach, large solution spaces would be limited sets of to allowed points (compare figure 1.3). I drew an analogy to the mechanical calculators shown in figure 3.29.

The rotary calculator on the left of 3.29 offers results for proportional problems for any possible input values, that is, at any possible rotation angle. The calculator on the right requires moving the mechanical monkey’s feet to two numbers on a linear scale at the bottom, which results in the monkey’s hands framing the product of the two numbers near the centre area of the calculator. This calculator can only multiply whole numbers as it offers no results when the monkey’s feet are
Figure 3.29: Two different mechanical calculators: Soviet nuclear contamination calculator and children’s multiplication calculator “Consul the educated monkey”.

located between the whole numbers on the scale at the bottom of the calculator. My thought was that a geometry rationalisation tool should maybe not allow the full kind of variety supported by the calculator on the left. Maybe it should be set up in such a way that only those points were accessible at which rationalisation was particularly efficient. Parametric bubble size variation would thus not be as flexible as it could in principle be, but it would be constrained to move between allowed points only, at which sets of different required space frame elements were small and allow for economic batch production.

Until this stage of the toolmaking study, I had not explicitly considered the use of tools made by others within my own toolmaking. The process of toolmaking is a self-referential process in that toolmakers may use tools previously developed by others in development of novel tools. In this process, tools may be used and appropriated in ways that were not originally intended by their makers. I now decided to integrate tools made by others as “building blocks” into my own work, thereby appropriating these tools in my own understanding of their use and applicability in order to develop a tool for generating visually irregular rationalised space grids.

The above-described attempt at achieving support for co-rationalisation in
irregular, foam-resembling space grid design was based on fixed bubble diameters, initially random cell locations and rule-based cell movement. In this approach, the resulting grid topology was not fixed at the outset of the rationalisation procedure. Now, hoping that a fixed topology would be more suitable for identifying preferred points along potential variation ranges, I considered a system with fixed cellular centre points and parametrically controlled bubble sizes. In what I initially approached as a quick and “sketchy” test in two dimensions only, I used Surface Evolver (Brakke [1992]) as a “geometric engine” to perform parametric transformations of cell sizes according to “virtual pressure” parameters associated with each cell. As a starting configuration, I defined a regular hexagonal grid of 100 seed points to generate a Voronoi diagram (see outer left grid of figure 3.30) of 100 hexagons. I randomly assigned a pressure parameter from a limited set of allowed pressure parameters to each cell in this grid and subsequently processed it using Surface Evolver. Limiting the set of possible pressures serves the same rationalisation purpose as limiting the number of possible bubble diameters in the unsuccessful approach described above. Some results of the procedure are shown in the three diagrams on the right of figure 3.30.

The foam structures were then evolved using Surface Evolver to adjust individual bubble sizes according to their pressure values as shown in the remaining three diagrams in figure 3.30. The structures shown at the upper right and the lower left use two and three different pressure values respectively, while the pressure differences are small enough to preserve the foams’ topologies. Struts are not rotated and merely undergo parallel translation. The negative consequence is that the two resulting structures have as few strut orientation angles as the

20This test and its outcomes are reported in greater detail in section 2.5 in Fischer [2007b]
original, un-evolved hexagonal lattice, which can be seen as limiting its visual irregularity. The positive effect however is that (at least in the demonstrated two-dimensional implementation) the entire structure contains only one single type of node with three 120° angles. The instance on the lower right uses three different pressure values with differences strong enough to alter the foam’s topological structure. Originally separate vertices now coincide in the same locations and some individual bubbles have collapsed. This results in varying strut rotation angles and hence in a higher degree of visual irregularity. The strut length variation is however quasi-random and shows little possibility for batch production while the increased number of different node types in this lattice also undercuts the intention of rationalisation. Using this method and Surface Evolver, it is in principle possible to model two- or three-dimensional space grids which are defined as two- or three-dimensional flat tori respectively. This was not yet implemented at that stage of the study. It would however have offered the positive effect that resulting structures would allow periodically repeated tiling to fill areas or volumes much larger than the initially generated “patch”. The repetition in such a structure should be modelled in such a way that an observer is unlikely to notice it. An advantage of this strategy would be that any number of strut and node elements counted in a patch would be definite and not increase with the addition of more identical patches either in whole or in part to increase the size of a structure.

An interesting possible application area for the two-dimensional lattices shown in the centre of figure 3.30 is the design of rationalised straight walls that are expressed to resemble liquid foam. This challenge was encountered during early attempts to rationalise the space grid structure for the Beijing National Swimming Centre. Straight walls can be described by first generating a two-dimensional lattice of the type shown. The depth of the wall space is produced by creating a copy of the
lattice, which is then rotated by $180^\circ$ and placed at an appropriate distance behind or in front of the original lattice. For each vertex in the original lattice there is a corresponding vertex point in the copied one. Connecting each two corresponding vertices with a crossbeam allows the formation straight walls of a single layer of bubbles as shown in figure 3.31. The two shown wall elevations are based on the upper right and the lower left lattices shown in image 3.30. In both cases, each strut length of the original grids is simply doubled in its copy and the orientation by $180^\circ$ results in a high degree of symmetry in the entire wall structure, which also reduces the number of different lengths of cross beams.

The distribution of different strut lengths contained in both walls is shown in figure 3.32. Light dots represent struts on the front and back panels and black dots represent the cross beams between the front and back panels. The distribution patterns for both walls show a tendency towards clustering of strut lengths (see figure 3.32). Clusters of sufficient numbers of struts and of sufficiently slim horizontal width are assumed to support cost-efficient batch production. Should panel members and cross beams be manufactured to be of the same type in a given application case, it would be possible to adjust the ratio between the two-dimensional bubble spacing and the wall depth in such as way that large clusters
3. TOOLMAKING STUDY

**Figure 3.31:** Elevations of two straight walls, both of a single layer of bubbles.

of panel member lengths coincide with large clusters of crossbeam length (in other words: to pile large columns of black dots in figure 3.32 on top of large columns of grey dots).

This approach was successful not only in terms of reducing the number of resulting strut lengths (see the obtained clustering effect in figure 3.32). The single initial node type with three struts meeting at angles of $120^\circ$ could be maintained with some sets of pressure parameters while an overall irregular visual appearance could be achieved (as shown in the two diagrams in the centre of figure 3.30).

Layering behind the resulting two-dimensional grid a rotated copy of itself and
connecting the two grids with a set of cross-beams resulted in a flat wall resembling liquid foam, which I believe is a very interesting response to the Beijing problem so far. This structure consists of convex, bubble-like polyhedra of different sizes, thus resembling liquid foam, it contains very few different strut lengths and node types and at the same time forms flat walls as required for the “Water Cube” in Beijing. The rotation and cropping shown in figure 1.21, which was used to increase visual irregularity but at the same time largely defeated the rationalising effect of using a Weaire-Phelan-like lattice is not necessary in this approach.

Trying to develop this approach into a co-rationalising tool I programmed a cursor-based tool to assemble Kelvin cells in three-dimensional space and to assign pressure parameters as in the two-dimensional test. This attempt was however not successful due to Surface Evolver’s handling of edges and surfaces. While the program processes edges (in two as well as in three dimensions) as defined in input data files, polygonal faces are triangulated by the introduction of a new vertex at the centre of the faces\textsuperscript{21}. Figure 3.34 shows a cluster of 15 Kelvin cells amongst which Surface Evolver maintains flat walls between any two adjacent cells of equal

\textsuperscript{21}This step is necessary to maintain the overall topology of processed structures and is analogous to the issue concerning over-constrained vertices discussed in section 3.5
pressure. On the outer surface of a cluster of cells, the triangulated surfaces are however smoothed similar to the way in which liquid foam would minimise its surface.

Figure 3.34: Exterior surface smoothing in Surface Evolver.

A similar effect occurs between adjacent cells with different pressures. This face triangulation by introduction of new vertices and the movement of new vertices out of the faces defined in the input data multiplies the number of contained strut lengths and node angles and contradicts the objectives of the intended geometry rationalisation. As shown in figure 3.35, this attempt to achieve the results obtained in two dimensions into three dimensions did not yield the expected rationalisation effect.

Figure 3.35: No clustering of strut lengths in three-dimensional grid structure.

Hence, I decided to remove Surface Evolver from the tool and to replace it with an algorithm of my own, which maintains flat surfaces between cells as well as on
the outer surface, involving no triangulation and smoothing on the exterior of cell clusters. This algorithm involved taking any two numerical pressure parameters assigned to any two adjacent cells (or one cell and a constant “atmospheric” pressure where no neighbour is present) and shifting their shared face towards the centre point of one or the other of the two cells. The effect would be that a cell with a pressure parameter greater than the pressure parameters assigned to all its surrounding cells would be larger in volume than the surrounding cells.

Rather than developing the required formula from scratch, I tried to think of some common thing that behaves in a comparable way and to look for a mathematical model of that common thing which I would then adapt and use. Thinking about a useful analogy from which I could model this algorithm I first thought about the balance scale. My initial thought was that two different weights resting on the two sides of the scales would cause the hand at the centre of the scales to give a reading of the relationship of both weights. This reading, i.e. the angle of the scale’s hand, should do what I wanted the film between two cells to do. I soon realised this was a wrong assumption: the side of the scales that carries the heavier weight always drops completely and the goal is to counter-balance it on the other side of the scales so as to move the hand into a neutral position. The next analogy that came to my mind after some thinking was the way voltage is divided along two resistors in serial connection (see figure 3.36).

The way in which partial voltages are impressed on the resistors according to their respective resistance offered a good analogy for what I had in mind. The algorithm by which surfaces between adjacent cells are placed is inspired by the below formula used to calculate partial voltages on a voltage divider.

\[ U_1 = \frac{U_{total} \cdot R_1}{R_1 + R_2} \]

In this formula \( U_1 \) and \( U_2 \) are partial voltages available at the serially connected
Figure 3.36: Algorithm for locating surfaces between adjacent bubbles based on voltage division.

Figure 3.36: Algorithm for locating surfaces between adjacent bubbles based on voltage division.

resistors $R_1$ and $R_2$ (see circuit diagram on the left of figure 3.36) whereas $U_{total}$ is the total voltage. As expressed in the below adapted formula accordingly, the dividing surface between the two cells shown in the diagram on the right of figure 3.36 is placed at distance $D_1$ from the centre of the upper bubble (or at distance $D_2$ from the centre of the lower bubble) while $D_{total}$ is the total distance between both bubbles’ centre points and $V_1$ and $V_2$ are the two pressure parameters assigned to the two bubbles.

$$D_1 = \frac{D_{total}}{2} + \frac{D_{total} \cdot V_2}{V_1 + V_2}$$

This algorithm results in a clustering of strut lengths similarly to the two-dimensional grids shown above in figures 3.30 and 3.32. With this algorithm implemented in the tool in place of the previously integrated Surface Evolver, I had now implemented a tool that allows the generation of rationalised irregular space grids resembling liquid foam based on three-dimensional clusters of spheres. Figure 3.37 shows the type of grid structure that can be generated with the tool.

This approach, supported by the tool, offers several opportunities for manipulating the compositional details and the external shape of an apparently irregular space grid structure while rationalising it simultaneously. Individual bubbles can be added, deleted or moved without disturbing the overall logic of the configuration. Moreover, the volumes of individual bubbles can be modified.
within the boundaries of available parameter sets to change form locally. For this purpose it would be necessary for the structure in question to undergo “evolution” after every manual change before the result would be visualised. These exploration processes can occur in a dialogue-based fashion while numerical surveillance of a designed structure, of its rationalisation and related cost implications can be made available to the designer automatically and continuously. This tool could not be used to address the rationalisation challenge of the Beijing National Swimming Centre as it cannot produce flat exteriors. A tool based on the two-dimensional test discussed above (see figure 3.30) would have been more suitable for the Beijing project.

Another effort undertaken at this stage of the toolmaking study aimed at implementing an algorithm to count the number of node types in a space grid structure. In order to evaluate the degree to which a space grid is rationalised, that is, in the context of this study, the degree to which repetition is introduced to harness the economic benefits of batch production, it is necessary to count the number of different strut lengths and of different node types contained in the structure. Different node types are nodes showing different patterns of strut connection angles. While counting the number of different strut lengths in irregular...
space frame systems is a straight-forward task, the counting of different contained node types requires the matching of patterns of strut connection angles and is thus more challenging. For this purpose, nodes can be thought of and modelled as small spheres and strut connection angles can be represented as points on the surfaces of the spheres. Relationships between angles must be compared and matched, and sub- and supersets of matches must be identified in a symmetry-sensitive way. By subset I mean a node which shows less strut connections than one in the superset while all its connection angles are amongst those found in the superset node. One challenge in this procedure is the necessity to compare node types in all possible combinations of rotations. This requires the relative comparison of patterns of connection angles, without fixed references such as constant poles or rotation planes about which nodes are translated between matching operations.

A problem similar to that of counting node types is encountered in determining spacecraft attitude control (see Shuster in Pisacane and Moore [1994], pp. 245-336). In order to determine the difference between current and target aircraft attitude, star configurations captured with an on-board camera can be matched against stored maps of star configurations (see figure 3.38).

Figure 3.38: Pattern matching between sets of points on spheres.

---

22I was given this reference and a general description of the matching algorithm by Michael Orion of Phoenix, Arizona on news:sci.math.research.
The matching of a captured image of the celestial sphere against a star map is, in terms of its basic geometry, comparable to the matching of strut connection angles on two space grid nodes. In their details, the two tasks however differ to the extent that many existing practical techniques from the aircraft navigation field, such as those dealing with noisy sensor readings, are not applicable to node type matching. Similarly, aircraft navigation can make use of data describing attitudes in the immediate or recent past to identify and prioritise likely candidates for matching rotations, which also does not apply to node matching. There seems to be little alternative to the exhaustive comparison of possible combinations of all possible rotations in order to determine the number of node types contained in a space grid.

The relative matching of angles on a node poses various challenges, one of which is symmetry. Two nodes can have identical sets of relative angles between strut connections but can nevertheless be different if they are mirror images of each other. However, there are pairs of nodes that can be identified as mirror images but are yet identical (this occurs for example in the simple case of two nodes, each of which with three strut connections in one plane separated by two equal angles). The development of a stable, generically applicable node-type counting algorithm in this study was ultimately hampered by issues of mathematical precision. Extensive vector transformations result in a buildup of imprecisions to the point of unreliability. Since vector angles and their transformations are independent of scale, it is not possible, to “scale” the calculation up to avoid imprecisions as may be possible in dealing with precision issues in counting strut lengths. While in applied construction the elasticity and flexibility of sufficiently long struts could be used to compensate for imprecisely matched node types, there would nevertheless still be a relationship between grid scale and node matching precision. Based on my
experience of developing a node counter of limited reliability, I assume it may be challenging to achieve a node counter of generic, scale-independent applicability as space grid scales and manufacturing tolerances may differ from project to project. While node-type counting initially appeared to be an abstract, theoretical problem concerned with “pure” vector mathematics, it increasingly appeared dependent on practical considerations and constraints as they are posed by given architectural projects.

In response to the secondary research question of whether the integration of non-design tools as “building blocks” into design tools constitutes a viable approach to toolmaking for design, observations made during this stage of the toolmaking study indicate a mixed answer. At this stage, I used and appropriated tools made by others, however in my own understanding of their use and applicability. While this approach to toolmaking for design provided inspiration, it was not necessarily conductive to my specific purposes. Replacing tools made by others with my own algorithm, I then succeeded in developing a tool for generating visually irregular rationalised space grids.

3.10 Tsunami Memorial Design Competition

The secondary research question investigated at this stage of the toolmaking study is: How does tool use as I anticipated it during the preceding toolmaking phase differ from actually using it in an applied context?

The primary research questions addressed by observations made during this stage of the toolmaking study are question 2: Can the development of digital design tools be justified after the abandonment of design methods? and question 3: Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers
Aiming to test the co-rationalisation approach in an applied design context, I applied the three-dimensional tool developed during the previous stage of the toolmaking study (see section 3.9) in a design project for the Tsunami Memorial Design Competition in Thailand in late 2005 together with Chris Bosse (PTW) and ARUP (Sydney) engineers Tristram Carfrae and Stuart Bull, who joined the design team shortly after work on the project started. Chris Bosse (PTW) and I chose the Tsunami Memorial Design Competition for this purpose after some weeks of casual email exchange during which the idea of a competition project took shape. The foam theme appeared to offer some potential for the design of a place to memorialise the victims, survivors and their helpers of the flood waves that followed an earthquake in the Indian Ocean in late 2004. Chris Bosse has a design portfolio of projects based on themes related to water, bubbles and foam and involving geometrical issues of close packing and minimal surfaces reaching back several years before his involvement in the Beijing National Swimming Centre. We were thus able to bundle architectural design experience, an international competition, a geometry generation and rationalisation tool relevant to the subject of the competition and, following the joining of the ARUP team, related structural engineering support.

The memorial was to be located on a coastal site with dense forest vegetation in Lamru National Park in Phangnga Province, southern Thailand. The competition brief included a description requesting entries to meet with the following major submission and program requirements:

1. A master plan of the site showing access, buildings and overall site concept in consideration of the natural features of the site.

See http://www.tsunamimemorial.or.th/program.htm
2. A memorial expression which may be an art piece or installation or landscape treatments or a separate structure or symbol or one that is integrated into the other facilities of the program.

3. A museum space.

4. A tourist and visitor space, including a small area for worshiping of all faiths.

5. A learning center for students and the public that may be separate or integrated into the tourist and visitor facilities.

Concrete work on the project began in mid-October 2005 in an afternoon-long meeting in Sydney during which basic conceptual ideas and possible modes for our collaboration were exchanged and discussed. Following this meeting, all further exchanges occurred on-line via email with Chris Bosse working in Sydney and myself in Hong Kong until our entry submission for competition stage I (see poster panels in figure F.1 and F.1) in early December 2005. During the Sydney meeting it was agreed that a group of foam-like pavilions should accommodate the program. It was also agreed that our roles as collaborators should be defined based on Chris Bosse’s experience as an architect and mine as a “geometry person” and toolmaker.

Chris Bosse led the development of the design proposal as the principal designer throughout our work on the project. While we agreed on most issues regarding the project, the exchange was somewhat hampered by our using of email as the primary means of communication. Design decisions rely to a great extent on intuitions and hunches that are difficult to communicate using text and images. Furthermore, it became apparent that we were interested in different aspects of the design proposal. My interest in geometry rationalisation resulted in a focus on clarity and precise specifications. Chris Bosse, in contrast, aimed to show a variety of possible bubble structures and utilised the tool and its output primarily to show a general capability for rationalisation while not committing to only one way of
expressing foam-like pavilions. Different to my interest in clarity, he, in my view, embraced ambiguous expressions of form throughout the design process and in the submitted competition entry (see section F).

Figure 3.39: Early sketches for Tsunami Memorial Design Competition by Chris Bosse.

The design proposal, as illustrated by concept sketches emailed by Chris Bosse shown in Figure 3.39, focused on a series of pavilions, mainly below tree height, scattered along a circular path accessible from the land side. A few pavilions of similar size were scattered in the sea near the site. While the soap bubble tool was made available to Chris Bosse for application during the design process, it was in the end operated only by myself, the toolmaker, to provide detailed structural models illustrating some of Chris Bosse’s design concepts. During the design process, needs emerged for supporting other types of foam, alternative skinning representations and alternative data exchange facilities. The latter two of these were accomplished, while a fundamental re-programming of the space grid topology proved too challenging within the available time frame.

The competition entry developed from this project did not win a prize. The winning entry of the first stage, however, also features several pavilions prominently displaying irregular space grids. In their comments on this submission at the first competition stage (see appendix F), the judges recommended that the space grid
structure needs “more clarity.” If this was addressed at the level of fabrication and construction economics, it would imply the necessity of a geometry rationalisation strategy. The further development of the winning entry, though, still does not seem to address the issue of geometry rationalisation. Its success indicates that attention to geometry rationalisation (and hence related toolmaking work) is not necessarily seen as being of key relevance in the evaluation of a preliminary design proposal. This corresponds to the common pattern of relegating rationalisation efforts beyond proposal stages as post-rationalisation strategies.

Figure 3.40: “Wet model” of bubble pavilions by Chris Bosse.

With respect to the secondary research question of whether and how tool use as I anticipated it during the preceding toolmaking phase differs from actually using it in a realistic context, observations made during this stage of the toolmaking study indicate considerable differences. The designer, who was free to use the tool by himself, left it to me to use it. He also suggested the implementation of new functions into the tool throughout the project. While my understanding that the tool’s purpose was that of finding unambiguous expressions of structures based
on one approach of geometry rationalisation, the designer utilised the availability of the tool to demonstrate a variety of options, accepting and even aiming for ambiguity.

### 3.11 Postgraduate design workshop

This stage of the toolmaking study directly addressed primary research question 3: Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers can?

With the design process for the Tsunami Memorial Design Competition, it was established that the tool I had developed was applicable to its intended purpose, in that it supported the development of space grids as part of an applied design process. The tool contributed to the proposal we submitted (the poster panels are reproduced in figures F.1 and F.2 in appendix F). At this point, it was however still unclear whether others would find the tool as supportive as I had. During previous stages of the toolmaking study, I was the only direct user of the tool and despite it being available to him, my collaborator did not use it during the work on the competition project. Following the Tsunami Memorial Design Competition I thus decided to make the tool available to others to observe their use of it in order to see how they would make use of the tool and how this would affect their design processes.

For the first time since the development of Zellkalkiil I therefore had to develop a design tool to a sufficient level of robustness to allow others to use it. I extended the tool to make it more generally applicable and suitable for use by others, and to also support packings not only of Kelvin cells foam but also of rhombic dodecahedra and of Weaire-Phelan-like cells, all with the “virtual pressure” based parametric
variability previously used for the *Tsunami Memorial Design Competition*\(^\text{24}\). I also extended the new version of the tool with the capability of translating grids of the three kinds into each other to allow for visual comparisons between them. Moreover, I improved data exchange facilities to other three-dimensional modelling packages and extended the user interface for greater clarity and overall reliability.

The improved tool was then applied in a postgraduate design studio workshop together with C. M. Herr (see Fischer and Herr [2007]), who used this opportunity to test a generative design tool she had developed as part of her postgraduate studies at the Department of Architecture at The University of Hong Kong. The workshop, titled *Tofu Cubes and Soap Bubbles*, was conducted during a five-day period in April 2006 at the Department of Architecture of the National Cheng Kung University (NCKU) in Tainan, Taiwan. Seventeen Masters and PhD level postgraduate students took part as members of five design groups. All groups were given the same, relatively open design brief, which asked for an extension to the NCKU campus to host additional space for the Department of Architecture. A list of functions and specifications was given to the students to provide a framework for those who might find a vague design brief too challenging. We nevertheless

\(^{24}\)I realised later that the parametric cell size variation strategy now supported by the tool could be regarded as a successful solution to the geometric problem presented in section 3.5 (see figure 3.27), which we had presented at the European Workshop on Computational Geometry 2003 at the University of Bonn, Germany. Following my attempt to achieve geometry rationalisation based on mobile cells (see section 3.9 and figure 3.27), it seems increasingly evident to me that the objective illustrated in figure 3.5 in section 3.5 cannot be achieved due to over-constraining. The solution, which was at this stage implemented for three different types of polyhedral packings in the tool (without intentionally addressing the original objective expressed in section 3.5) is possible due to fixed cell centre points.
encouraged deviations from and re-interpretations of the brief according to the students’ interests and no more than preliminary, conceptual results were expected within the short time frame of the workshop. We asked the students to use both our tools named the “tofu cube tool” (by C. M. Herr) and the “soap bubble tool” (the extended software I previously used for the Tsunami Memorial Design Competition as reported in section 3.10)\(^{25}\). Corresponding to the two different purposes for which our generative design tools were developed, design proposals for the new building were expected to consist of concrete building masses and glazed space frame structures. Students were free to utilise any resources and tools they wished to use in addition to our two tools in their design processes.

The tofu cube tool is a 3ds MAX\(^{\text{R}}\) script, accessible through the graphical user interface of a 3ds MAX\(^{\text{R}}\) utility. The script enables users to generate complex assemblies of white box shapes, hence the name of the software. Four basic materials are available to assign to box shapes in a 3ds MAX\(^{\text{R}}\) scene: “matter”, “void”, “context” and “neutral”. Objects with any of these materials may then be repeatedly transformed geometrically either manually or through a number of predefined functions (see figure H.3 in appendix H). Intended for bottom-up form generation, this tool supports processes that oftentimes lead to unexpected results. The tofu cube tool is one of a series of test implementations exploring the potential of cellular automata as architectural design support tools (see Herr and Kvan [2007]). Based on observations made during previous tool implementations (see ibid.), the tofu cube tool was developed to offer a range of easily accessible

\(^{25}\)During the workshop and in the following published report (see Fischer and Herr [2007]) we called the tofu cube tool “Tofu Automata Generator” and the soap bubble tool “Soap-Bubble Truss Co-Rationaliser”. We stopped using the original names not only because they are very long but also because of the emphasis on the word “co-rationalisation”. This issue is discussed in detail in section 4.2.5.
3. TOOLMAKING STUDY

predefined functions to generate complex forms based on given box shapes.

The soap bubble tool is a VBA macro script implemented in MS Excel® and Rhino3D® with a graphical user interface (see figure H.1 in appendix H). It allows the editing of close-packed sphere structures or “bubbles” and their translation into corresponding space frame structures. Individual bubbles can be assigned virtual “pressure” parameters to obtain bubbles of different sizes within the same three-dimensional cluster analogously to the two-dimensional lattices shown in figure 3.30. The tool supports lattices based on packings of three different space-filling convex polyhedra: rhombic dodecahedra, truncated octahedra and Weaire-Phelan-like cells (see appendix E).

To explore whether tools for generative design can be purposefully passed from the context of their production to other contexts, Christiane Herr and myself identified three aspects in terms of which tool applications could diverge from toolmaking intentions. These three aspects, briefly outlined in this section, were used as primary measuring devices to investigate the research question. The first of these aspects is the distinction between “top-down” and “bottom-up” oriented design processes, which has been used to describe and classify generative design approaches based on their theoretical and methodological underpinnings. Schmitt [1993], pp. 42-45 gives a basic discussion of both approaches. More recent examples include Chase [2005] and Scheurer [2005] (see also section 1.2.5). In the case of top-down orientation, initially defined expressive gestures are iteratively differentiated to fulfill some additional design criteria. Parametric geometry variation is an example. In the case of bottom-up design, configurational rules are iteratively applied to generate forms that are initially hard to predict. Cellular automata-based composition of form is an example. We regard one of our tools as bottom-up oriented and the other one as top-down oriented. To assess whether this distinction
has an effect on the use of the tools, we asked the workshop participants to rate (on a scale of 1 to 7) whether they used the tools primarily to realise ideas they had before they started using the tool (1) or to generate ideas through exploration and experimentation (7). The questionnaire sheets are shown in figures H.4 and H.5 in appendix H, and collected data in appendix I.

The second aspect we identified relevant as for tool use in design is that of how a joint creative effort is orchestrated within a group of designers. Kvan [2000] draws a distinction between close-coupled collaboration and loose-coupled co-operation of designers (see also section 1.2.5). He argues: “A loose-coupled design process requires a very much different set of tools and conditions to be successful than a close-coupled one” (ibid., p. 415). Inverting this argument, it could be expected that there would be a distinct type of tool appropriate for each kind of collaboration. The GUI interfaces of both of our generative design tools occupy less than 1/8 of the screen surface and are operated sequentially via keyboard and mouse. Largely determined by our intentions at earlier stages of the tools’ development histories, their interfaces favor individual use and hence processes involving loosely-coupled design co-operation. By observing the degree to which tools are used in loose or close coupling, we aimed to obtain indications regarding the degree to which toolmaking intentions shape tool use. For this purpose we asked workshop participants if they used the tools individually, working closely with group members or both.

The third aspect relevant to us concerns Glanville’s [1992] and [1994a] distinction of tools from media, as discussed in section 1.2.6. As a tool, the computer can responsively follow user instructions and intentions preconceived by the toolmaker. As a medium, however, the computer can lead the user to results that were neither intended nor expected by the toolmaker or the designer, and
thus plays an interactive role in the design process. To examine if and in what ways the two tools might have acted as media during the workshop, we observed the design processes and examined the digital material generated during the workshop, seeking types of results that were different from the conceptions we had when making the tools.

During the workshop we provided tutorial support and commented on the resulting design work in the context of a final critique. Apart from that, we observed the five groups’ design processes at four different levels: We took field notes during the tutorial interaction, focusing on events and observations that seemed interesting, unexpected or otherwise relevant with respect to differences between our and the students’ way of using the digital tools. Students maintained self-observation notes, recording times and types of activities involved, respective environments, resources used and related remarks. Before the final critique, we conducted a questionnaire with individual students (see figures H.4 and H.5 in appendix H and appendix I) and after the final critique we collected all digital material produced by all groups during the workshop. All groups designed, presented and discussed interesting and original designs on the 5th and final day of the workshop (see figure 3.41).

Fifteen participants used the tofu cube tool and seventeen used the soap bubble tool. While we regarded the tofu cube tool as a bottom-up design tool and the soap bubble tool as a top-down design tool, the participants rated their uses on average 5.1 in the case of the tofu cube tool and 5.5 in the case of the soap bubble tool (1=I used the tool to realise ideas I had from the beginning of the workshop, 7=I used it to explore and experiment). In addition to these inconclusive numbers, our observations during the workshop also did not suggest that clearly identifiable top-down or bottom-up design processes were carried out. The categories of top-
down and bottom-up designing did not seem to play a role in the students’ design thinking, which was apparently not very concerned with the tools we provided but with aspects of the design task that lay outside the tools, such as site access, urban context and so forth. According to the quantitative measurement of our preconceived distinction between top-down and bottom-up design approaches, both tools were applied in predominantly bottom-up ways, the one we thought of as being more top-down oriented even more so than the one we regarded as being bottom-up oriented.

According to the questionnaire data (see appendix I), both the soap bubble tool and the tofu cube tool gave students surprising results, but students incorporated only a selection of these into their design outcomes. This indicates that students pursued a certain direction in their design processes and did not completely rely on unanticipated outcomes generated by the software. The answers given to questionnaire items 5 and 19 (Does your final tofu cube / soap bubble tool outcome match your initial design ideas?) and 6 and 20 (Did using the cube / soap bubble inspire you and help to explore new design ideas?) indicate that while
students were often surprised by the outcomes generated with the software tools, these results were however not necessarily rated as very inspiring or essential in exploring new design ideas. When asked which type of bubble packing gave them the visually most interesting results, almost all students agreed on the Weaire-Phelan-like cells.

While we regard both tools as supporting single-user dialogues best and favoring loosely-coupled design processes, only one user of the tofu cube tool always used it by herself, four always used it working closely with group members and eight went back and forth between the two modes. None of the users of the soap bubble tool used it entirely by themselves, seven used it working closely with group members and eight worked in both modes. Students using both tools either developed their initial ideas on paper and in close collaboration with other group members, or only relied on the software. None of the students using the soap bubble tool and only two students using the tofu cube tool sketched separately from other group members to develop ideas, as would be expected in conventional design processes.

Throughout the workshop, students’ interactions with the software were tracked (see table I.1 in appendix I for a sample of the logged data). We measured the frequency of function use within the soap bubble tool and the tofu cube tool with the intention of capturing the nature of students’ interaction with the software. As students frequently found ways to use the software in unintended ways, as discussed below, outcomes of this quantitative measuring can only be taken as indicators of the nature of students’ design processes.

The tracking of students’ interactions with the soap bubble tool also indicates use patterns that differ from the perceived usefulness ratings. The chart in figure 3.42 shows that the most frequently used feature of the soap bubble tool was the “cursor”, with the “volumes” feature used only half as often. The “boxfiller”
was used even less frequently. The “storages” feature was used only occasionally, though students perceived this feature as very useful.

One reason for this pattern could be that students might have felt anxious to lose bubble configurations they were working on, and relied on the storage function to provide a backup. While not used frequently during the design process, the availability of a backup if needed might have been important to allow students to explore without being afraid of losing data. On the other hand, this pattern could also indicate that while being aware of the option of reverting to previous stages of their work, students instead preferred to go along with potentially surprising results of the software\textsuperscript{26}. Overall, it seems that the quantitative data

\textsuperscript{26}Similar to the “storages” function in the soap bubble tool, the tofu cube tool also featured an “undo” function which could be used alternatively to the generic “undo” function of 3ds MAX\textsuperscript{\textregistered}. The difference between the two could however not be tracked reliably in the data logging process, such that it was not featured in
collected during the workshop contributed little to our understanding of the students’ use of the digital design tools. With the students’ understanding and use of both tools being highly individual, and quantitative measurement being limited to preconceived categories, the interpretation of collected data in terms of causal relationships must remain at a speculative level. Qualitative data collected over the course of the workshop through straight-forward observations and the recording of “field notes” as well as conversations during tutorials and during final presentations yielded more valuable insights.

There were for example at least four cases in which our tools were used in ways we did not anticipate (see figure 3.43 for examples). Group 1 was inspired by forms generated with the tofu cube tool, but then modelled a related form by other means. Group 5 was conceptually inspired by geometrical translations observed in the same tool, then proceeded to design an adaptive kinetic library system without applying either tool. Group 2 used the soap bubble tool to generate an irregular canopy structure in conjunction with other tools and used the tofu cube tool, which is intended for non-deterministic form generation, to generate a model of the project’s site context. Group 4 found previously ignored ways to obtain surface “eversions” – polyhedral intersections – by exploiting extreme parametric relationships, and interpreted the results as furniture and similar interior design elements. Users also requested additional software features such as a formZ® export filter for the soap bubble tool, which we were able to provide during the workshop, as well as ways to multiply soap bubble clusters generated with the soap bubble tool in the tofu cube tool, which was impossible to achieve before the end of the workshop.

Another example of unanticipated tool use is shown in figure 3.44. It shows the questionnaire.
part of a slide from the final presentation of group 4. Showing this image, the group members explained that they had used the tofu cube tool to explore different levels of visual complexity and found out that they were interested neither in low degree of visual complexity shown on the left of the diagram, nor in the great complexity as shown on the right, but in a medium level of complexity as shown in the middle of the diagram. The group apparently used what was developed as a form-generating tool to think about and make decisions about their design goals. Moreover, the group then used the tool to communicate this strategy to others. According to Glanville’s [1992], [1994a] distinction between tools and media for designing (see section 1.2.6) the software was used as a medium. This suggests that the developer’s understanding of design software may not be very relevant and that the role software plays in designing is primarily defined through the process of application by the users.
3. TOOLMAKING STUDY

Overall, our observations can only be limited as we have tested only two generative design tools in only one workshop. Assuming that with increasing mastery of a tool, the modes of its use can change and adapt over time, it might very well be that the five-day period of the workshop was not long enough for this adaptation to become effective. Nevertheless, findings in all three examined aspects are counter-intuitive to the idea of passing generative design tools with constrained purposes from the context of their production to other contexts. In this respect the application of tools to designing seems to differ significantly from the application of other tools such as hammers and bicycles.

Apparently, the purposeful generalisation of a design tool requires some coincidence between the user’s design intentions and the toolmaker’s understanding of the user’s intentions. In generative design, however, as shown in the study presented here, this seems to not always be the case. If a tool proves useful to a user for a purpose that was not considered by its toolmaker, then this usefulness must be described as either accidental\textsuperscript{27} or as a result of abuse. The word “abuse” is used here in reference to Glanville’s \cite{Glanville1992} and \cite{Glanville1994} use of the term. Burry \cite{Burry2003} refers to the same phenomenon using the term “appropriation”. The difference between the two terms illustrates the differences of two viewpoints: that of a toolmaker who may have an interest in exercising (restricting) control and that of a tool user, whose interest is in making new meaning.

From the perspective of an outside observer then, any other resource, being most likely equally unintended for the given design intention, could have proven just

\textsuperscript{27}Accidental usefulness may be regarded as a desirable key to novelty generation and be actively sought by designers. The accident has been described by Paul Virilio as “part of the production process” and as “an element of rationality”. See Virilio interviewed by Ruby \cite{Virilio1998}, p. 32.
as useful to the user as these tools that became useful through accident or abuse. From the perspective of the tool user, as shown in our study, the act of choosing a tool, as well as new concepts that emerge through its use, have an actively shaping influence on the design process. The task of the digital design toolmaker becomes complicated and compromised. If some aspects of the usefulness of one’s products depends on accident or abuse, then there is little to be learned from user observation in the way designers learn from observing the use of hammers or bicycles, whose makers and users we believe have some coinciding understanding of their use (see also Fischer [2008]).

Seeking a possible way out, toolmakers might look towards the support of tasks that are sufficiently “pure”, closed-ended and tamed (see Rittel and Webber [1973]), in the way in which CAD packages support tasks of pure geometry. In this case, however, it could be argued that software acts as a tool and not as a medium in Glanville’s [1992] and [1994a] (see section 1.2.6) sense and hence does not support the design process itself.

Moreover, generative design tools usually appear too “biased” towards specific types of forms and processes to qualify for this approach. The chances for providing generative design support in a directed manner by means of toolmaking appear slim - unless, possibly, at least one of the following two conditions is fulfilled:

1. Makers and users of design tools “conspire”, (see Rittel in Cross [1984], pp. 326-327 and section 1.2.4), i.e. operate within the same design context (project) and share a similar view of the tasks at hand. In the extreme case, the sufficiently skilled designer would make her or his own tools as described by Ceccato [1999].

2. Toolmakers find modes of design support in which tools are used as they intend, yet support the systematic generation of new thoughts, ideas and
understandings in the process of their use (see Glanville [1992]), possibly by exploiting ambiguity in conceptual presentations in the way sketches have been described to do (see Gross and Do [1996]).

Overall, observations made during this stage of the toolmaking study thus indicate a negative answer to the primary research question investigated at this stage, of whether digital design tools can be generalised from the contexts of their production to other contexts and to other users in the way in which non-design tools like scissors and bulldozers can. While designers may apply digital design tools offered by others, they either put them to the kind of use that Cross [1977], pp. 140 ff. (see also section 1.2.5) refers to as “hackwork” or they appropriate them and put them to uses unanticipated by the toolmaker.

3.12 Summary

Investigating the secondary research questions at each stage of the toolmaking study yielded the following results. Drawing on observations made during the first stage, it seems that the computer-based simulation of the processes described by developmental biology do not offer useful guidelines for the development of digital tools for designing. At the second stage, students taking a fresh look at Zellkalkül could contribute only limited input for further steps in the toolmaking process and did not succeed in applying the tool in ways that addressed applied design challenges. The third stage of the toolmaking study investigated whether autonomous behaviour of robotic elements could express patterns of an underground traffic network and indicated a negative answer to this question. During the fourth stage of the toolmaking study, investigating the secondary research question of which geometric principle allowing parametric shape control
could be incorporated into packings of rhombo-dodecahedral cells, I implemented and demonstrated one of a number of potential candidates. Some geometric elements are over-constrained in this solution, somewhat limiting parametric flexibility. These results do not seem to offer a practical solution to the problem of jagged surfaces and further complicate scripting efforts. At the fifth stage of the toolmaking study, I was able to derive a visually irregular spatial tessellation from the previously explored packing of rhombo-dodecahedral cells. During the sixth stage of the toolmaking study, I developed a proposal for a potential solution in response to the secondary research question of whether the geometry rationalisation approach developed during the previous stage of the toolmaking study could be modified to form a structural façade system that casts tree-like shadows. At this stage of the toolmaking study, my focus was less on toolmaking than on providing specific design proposals. Observations made during the seventh stage suggest a mixed answer to the question of whether the approaches described as pre-, post and co-rationalisation offer design procedures through which rationalised space grid structures resembling liquid foam could be achieved. While it is possible to categorise all geometry rationalisation work undertaken in the study so far as pre- or post-rationalised, exclusive categorisation of some cases into just one approach seems difficult. Furthermore, such categorisations seems possible only in retrospect. In stage eight of the toolmaking study, I integrated tools made by others as “building blocks” into my own work, thereby appropriating these tools in my own understanding of their use and applicability. While this approach to toolmaking for design provided inspiration, it was not conductive to my specific purposes. Replacing tools made by others with my own algorithm, I then succeeded in developing a tool for generating visually irregular rationalised space grids. Throughout this stage, I further explored the
notion of co-rationalisation. According to observations made during this stage, the three approaches of pre-, co- and post-rationalisation do not seem to offer guiding principles by which designers could proceed in rationalising architectural geometry. Observations made during the ninth stage of the toolmaking study further suggest considerable differences between tool use as I anticipated it during the preceding toolmaking phase and its actual use in a realistic context. While my understanding of the tool’s purpose was that of finding unambiguous expressions of structures based on one approach of geometry rationalisation, the principal designer with whom I collaborated during the Tsunami Memorial Design Competition utilised the availability of the tool to demonstrate a variety of options, accepting and even aiming for ambiguity.

Following the toolmaking study, the primary research questions can be answered preliminarily on a “yes-or-no” basis as follows:

The answer to primary research question 1 (Can natural or biological analogies guide the development of digital design tools?) is, as far as observations made during the toolmaking study are concerned, negative in cases where principles explaining natural phenomena are followed strictly, aiming to reproduce these phenomena (as in the case of cellular development and Zellkalkül). Primary research question 1 can however be answered positively where observations in nature are used as informal inspiration (such as using soap bubbles as inspiration for building form).

The answer to primary research question 2 (Can the development of digital design tools be justified after the abandonment of design methods?) is mostly negative with exceptions where tools are developed and applied within the same context.

The answer to primary research question 3 (Can digital design tools be gener-
alised from the contexts of their production to other contexts and to other users in
the way non-design tools like scissors and bulldozers can?) is negative.

The answer to primary research question 4 (Can the concepts of pre-, co- and
post-rationalisation offer guiding principles for the development of digital design
tools?) is mostly negative in the case of pre- and post-rationalisation while the
notion of co-rationalisation has apparently guided digital design toolmaking with
a successful outcome.
This chapter reflects on the toolmaking study presented in the previous chapter and establishes explanations for observations made during the toolmaking study within a second-order cybernetic framework. This chapter then proceeds to outline a personal model of novelty and knowledge generation, which accommodates the established explanations as well as some concepts regarding novelty and knowledge generation in science and design previously proposed by others.
4.1 Introduction

This chapter offers a reflective analysis of observations I made during the tool-making study reported in chapter 3. Section 4.2 uses second-order cybernetics (see section 1.5.4) as a theoretical framework. It discusses seven aspects of the theory and applies them as explanatory principles to observations I made while making and applying digital tools for designing. The purpose is to further examine the primary research questions (posed at the end of chapter 1), for which answers have so far been indicated on a preliminary “yes-or-no basis” in chapter 3, on a more substantiated “why basis”. Section 4.4 presents a personal descriptive model of designing that accommodates the explanations for my toolmaking experiences based on second-order cybernetics given in section 4.2 as well as some concepts regarding novelty and knowledge generation in science and design that have been covered in the review of literature and work of others in chapter 1. Section 4.3 outlines the findings of obtained in this thesis regarding the primary research questions listed at the end of chapter 1. Section 4.5 then relates the descriptive model of designing back to overarching questions posed in the introductory chapter 0. The following section 4.6 summarises and concludes the work presented in this thesis. The chapter then outlines the contributions offered by this thesis in section 4.7 and concludes with an outlook on future work based on this thesis in section 4.8.

4.2 Aspects of reflection

Following the postgraduate design workshop, the similarities, differences and relations between two observations I made during the toolmaking study, began to absorb my attention:
1. Programming cells in Zellkalkül from the inside-out perspective could not be applied to support designing.

2. Toolmaking for designing did not seem to work outside of applied projects, but within the contexts of applied projects, which, in some sense, also represents an inside-out perspective.

I did initially not know whether thinking about both observations in terms of a distinction between outside-in and inside-out perspectives was reasonable at all. What separates outsides from insides and what do both observations have in common such that they can be compared? I began drawing analogies to other scenarios, which I thought were comparable in the same sense. A soccer player, for instance, has an effect on her or his match and so does her or his coach. The player is inside of “the game”, the coach is outside. An airline pilot has an effect on the course of her or his plane, just as the air traffic controller in charge for the airspace the plane is in. The pilot is inside the plane, the traffic controller is outside. What is it essentially that distinguishes the role of the soccer player from that of the coach or that of the pilot from that of the air traffic controller? Could an answer to this question be expected to explain why, in the case of Zellkalkül, the inside-out perspective could not relate to designing while in the other case, it was a requirement for successful designing?

With these questions in mind, I found reading on second-order cybernetics helpful. I had read discussions of the theory before and found its positions clear and agreeable the first time already. Now, re-reading these discussions and relating them to observations I made during the toolmaking study, the theory took on more meaning. The practical experiences gathered during the toolmaking study and the theoretical principles put forward by second-order cybernetics seemed to reflect each other, while observations made in the toolmaking study exemplified various
aspects of second-order cybernetics. Second-order cybernetics, in turn, offered explanatory principles for the observations I made.

In the following sections 4.2.1 through 4.2.7, I discuss results and observations of the toolmaking study in light of seven aspects of second-order cybernetics, which I believe are particularly interesting when applied to the observations as explanatory principles. This discussion also addresses the primary research questions posed at the end of chapter 1 at an explanatory and more theoretical level than they could be addressed in the immediate context of each stage of the toolmaking study in chapter 3.

4.2.1 Conversation and novelty generation

Second-order cybernetics describes the design process as a conversation process (see Glanville [1999] and section 1.5.4). A design conversation takes place between two sides, each of which can be described in terms of the “states” by which it is able to accommodate the respective other side’s articulations. This is possible either by having predefined suitable states available or by constructing new ones. The states represent either side’s knowledge. Based on second-order cybernetic theory of designing put forward by Glanville [1999], pp. 9-12, figure 4.1 illustrates different design conversation scenarios that I found myself in during the toolmaking study: with others, with imagined others, with a computer (software) and with physical objects.

One can have design conversations with someone else, such as a collaborating designer (see upper left of figure 4.1) as in the case of my participation in the project for the Tsunami Memorial Design Competition discussed in section 3.10.

One can also have design conversations with oneself, taking on different roles and perspectives and thus simulating conversations with others (see upper right
4. REFLECTIVE ANALYSIS

Figure 4.1: Design conversations.

of figure 4.1). This was the case when I developed and tried to apply Zellkalkül, initially enthusiastically but then growing increasingly skeptical when the hoped-for results failed to appear. I eventually decided to abandon this approach to supporting design after receiving sceptical but vague comments when presenting my work (see section 3.4) and trying to imagine and to understand what it was that prompted the commentators’ scepticism.

One can also have design conversations with computer software (see lower left of figure 4.1). It is a basic assumption underlying this thesis that software tools are capable of supporting the generation of novelty and knowledge by offering the user output that is not immediately understood but which can be accommodated by the user’s construction of novel knowledge. This was likely the case when participants of the postgraduate design workshop, using the soap bubble tool, utilised unintended effects in parametric geometry variation to suggest interiors and furniture.

One can also have design conversations with things in the physical world (see
lower right of figure 4.1). This was the case when I used nodes with tetrahedral angles and straight bamboo sticks to model space grids based on packed truncated octahedra (see section 3.8 and figure 3.26). In order to force the bamboo sticks to connect with the nodes, I had to bend them, which resulted in edge curvatures that are not only similar to those observed in corresponding liquid foam structures, but also pre-stresses this grid configuration and gives it interesting structural properties.

During the toolmaking study, I increasingly regarded the outcomes of designing (including tools for designing) as articulations within conversations. Similarly to other kinds of articulations, such as those in language or music, there was little point in articulating design ideas without a receiver willing to pay attention. In this sense, making design tools for no applied design problem is not much different from making them for no particular user. In order for design conversations with others or with oneself to facilitate designing, both parties must be ready to accommodate what they do not yet know, and make new knowledge from it. Those groups who, in their interactions with the soap bubble tool during the postgraduate design workshop, appropriated the tools for new uses that I as the toolmaker (and the students themselves before the tool interaction) did not foresee, took advantage of unexpected feedback and utilised it in creative ways. This type of software use is a good example of digitally supported novelty generation and, according to Glanville [1992], [1994a], qualifies as a design medium as discussed in section 1.2.6.

It is also interesting to reflect on the recurring authorship question in collaborations between architects and engineers discussed in section 1.4.3 in the context of design as conversation and the notion of “betweenness” described by Glanville[2000], p. 4 (see also section 1.5.4). There is no single “originator”
of novelty where novelty is generated in design conversations between multiple participants, as creativity exists “between” those who are involved. The concept can also be applied to the interaction of entire disciplines (represented by the disciplines’ members and their articulations). This is how I understand Evans’ [1995], p. xxvi (see section 1.4.3) notion that architecture is not passively consuming knowledge from geometry but that the two disciplines consume knowledge of and contribute knowledge to each other.

4.2.2 Inside-out and outside-in views

The aspect discussed in this section addresses directly the questions regarding inside-out and outside-in relationships posed above at the beginning of section 4.2. According to Glanville [1997], the boundary that distinguishes an inside from an outside in the sense discussed here is constituted through the act of observing and what is being bounded is generally referred to as a “system”. By deciding whether something observed is stable (that is, whether it is aiming for a goal that is assumed to be stable) or not, an observer places herself or himself inside or outside of a system. This allows for four different possible relationships between observers and systems which are described by second-order cybernetics (see Glanville [1997] and section 1.5.4). The observer can be

1. outside of the system, looking outwards,
2. outside of the system, looking inwards,
3. inside of the system looking outwards or
4. inside the system looking inwards.

Glanville [1997], p. 9 notes that the condition arising from scenario 4 is concealed from outside observers and “we cannot, as outsiders, speak of this” (ibid., note 8). The remaining three, if applied to design research, seem to correspond to the
more general cases of the three modes of design research described by Frayling [1993/94], p. 5, Findeli [1999], p. 2 and Downton [2003], p. 2 (see section 1.2.4). Research about design corresponds to relationship scenario 1, research for design corresponds to relationship scenario 2 and research through design corresponds to relationship scenario 3. On this basis, the different possible relationships between observers and systems provide useful models for the different investigative modes I used in the toolmaking study outlined in chapter 2 and reported in detail in chapter 3. Designing occurs when the designer is in a conversational system with his means of designing, looking inside (relationship scenario 4), and at the same time, in another conversation, looking outside, aiming for something as-yet unknown, which is determined (made) through the process of designing (relationship scenario 3). The designer thus has two conversations at the same time: One conversation is aiming for the as-yet unknown, while the other conversation is happening in the system in which the designer is with the means (i.e. tools) by which he or she aims to reach the as-yet unknown. The direction towards the as-yet unknown changes together with the designer’s understanding of the design problem throughout the design process. 

This helps in explaining why design toolmaking in the laboratory, away from applied designing, does not seem to work well and why it works in applied designing (as discussed in section 3.10) and in advanced design practices such as Gehry Partners and Foster and Partners, who have established in-house toolmaking support that develops and adapts tools in direct response to the offices’ design projects. In teams in which digital design toolmakers support designers while designing, the designers have one design conversation aiming at the as-yet unknown (which, once it has been reached, is turned into new knowledge that can manifest itself in the physical world and is perceived as novel). The designers can have another, internal
conversation with the design toolmakers, who, with their work, are a means of support for the designers to attain the as yet unknown. A single person can be the designer and the design toolmaker at the same time, so long as this person manages to play both roles, allowing both conversations to happen.

The above also helps in explaining the difficulties I experienced in trying to program cells to develop and express design outcomes autonomously. When aiming to achieve “automated form finding” by simulating processes of developmental biology (see sections 3.2, 3.3 and 3.4), the cells could not produce form of relevance to applied design problems. Cells in Zellkalkül were equipped with internal means to perform functions I modelled after processes described by cellular biology (see appendix B). Performance of these functions did not correspond to the inside-in view of designing as all cellular code was provided from the Zellkalkül user, and cells had no means to externally as happens in the inside-out view of designing. Moreover, Zellkalkül never performed negotiations between mismatching states resulting in novelty as described in 4.2.1. Whether computer software can perform such negotiations at all seems to be an open question (with respect to which I tend to be pessimistic).

After having developed design tools for a while (at some point in the second half of the toolmaking study), I eventually stepped out of the toolmaking effort by reading about toolmaking and tool use in animals. I thus began to take the outside-in perspective towards toolmaking, which resulted in the overall structure and argument of this thesis. This is when I realised that there is not much of interest in tools by themselves. There is nothing exciting about the sticks and stones used by birds, primates and other species (see figure 1.1). What is interesting in animal tool use is what happens between the animal and the tool. Based on the work presented in this thesis, I believe the same applies to digital design tools. There is not much of
interest in digital design tools by themselves unless somebody uses them to make new things.

4.2.3 Analogue and digital knowledge

Designing produces encodings of knowledge, be it in the form of language, in visual, physical or other forms of expressions. Encodings produced in the toolmaking study described in chapter 3 include images, text, poster panels, physical models and computer software. Knowledge can hence exist in analogue and digital forms. In order to encode (digitise) knowledge, one must map continuous perceptions onto discontinuous representations such as words, numbers, materials, forms and so forth. All of these representations can at some level be understood as analogue but one can decide to arbitrarily assign them to, and to evaluate them in terms of, made-up representational codes. In this case they become digital symbols within digital (or notational) systems that, if not used playfully or dismissed entirely, are barely negotiable when applied in design contexts. One factor that seems to increase the degree to which design outcomes are perceived digitally is the textual content of design presentations (verbal, poster texts, image captions etc.) by which the less digital contents (images, models etc.) are explained, contextualised, justified and otherwise supported. Digitality can thus shown to be in the interest of the sender who wishes to achieve control over the receiver’s understanding of what is communicated (see section 4.2.6). bing a means for control, digitality prohibits the making of new meaning and knowledge, and thus designing. It must be noted in this context that scholarly articulations in the field of design research have a high degree of digital content (written words, empirical data, etc.).

As Glanville noted (in a personal conversation), the digital allows for control
and the analogue allows for variety\(^1\). In order to achieve control in something continuous, it is necessary to introduce defined “control points” that can be manipulated and evaluated in unambiguous terms. This is encoding. Symbols of a formal language (or of natural language that is used in formal ways) allow control in communication in a similar manner as the parameters by which ruled surfaces can be described allow for control in geometric design exploration. Digital and analogue knowledge (see Downton [2003], p. 65-66 and also appendix M) have different advantages and disadvantages in designing. Goel [1995], p. 166 uses the examples of a seismograph readout and a phone number to illustrate the difference between the two kinds of knowledge. Wilden [1980] describes the mapping of (analogue) continuity onto (digital) discontinuity as an epistemological necessity (ibid., p. 166) but notes that the analogue is “full” while the digital is “full of holes” (ibid., p. 192). This difference between analogue and digital knowledge is also what Dorst [2003], p. 76 refers to in describing the dilemma between “realism” and “clarity” (see section 1.2.4). Figure 4.2 shows how the continuous perception of some phenomenon (represented by the smooth gradient in the top bar) is mapped in two different ways (maybe by two different observers or by one observer at different times) onto discontinuous representations (illustrated by the two lower bars)\(^2\).

The original perception represented in the upper bar has greater, fuller “realism”, which is difficult to communicate entirely using a symbolic code or formal

---

\(^1\)In my understanding, this is the difference between the terms “high definition” and “high fidelity”.

\(^2\)This is the kind of mapping that happens when a participant in a design conversation perceives an utterance of the other participant without immediately understanding it, but then accommodating it using existing (and thereby making new) knowledge, as described in section 4.2.1.
language. The representations in the lower two bars allow for greater control ("clarity") in symbolic communication but the richness and what Dorst (ibid., p. 76) describes as "realism" of the original perception is lost. This difference in quality between analogue and digital communication and knowledge became apparent during the toolmaking study, for instance at the Tsunami Memorial Design Competition stage (see section 3.10). My interest in geometry rationalisation, coded design procedures and quantitative specification proved to be of little consequence in the outcome of the competition, whereas Chris Bosse made seemingly deliberate use of ambiguity as it is apparent in the model shown in figure 3.40 and its ability to stimulate the observer’s imagination.

The loss of realism in digital systems may be of negative consequences if essential qualities of the original perception are sacrificed. It also has negative consequences where communication requires the absence of control so as to produce novelty, as is the case in design. Digital representation may not bring negative consequences where the most important objective is simply to communicate precisely and when the loss of realism does not matter to the observer. This is the case in building
fabrication and it accounts for the success of digital tools in design manufacturing.

The representation and control of continuous phenomena using digital encodings lessens (possibly crucial) qualities of what is represented. But there are cases when this does not necessarily matter. An engineer may grow unaware of the tolerance level to which his or her company manufactures space frame components (see footnote 11 in section 3.6), just as a music lover may be satisfied with CD quality audio or a computer user may find a display with 24-bit colour depth sufficient to view digital photographs without experiencing flawed perceptions. There are levels at which observers can distinguish distinct states and there are thresholds beyond which observers fail to distinguish distinct states. In the latter case, the negative effects of digital encoding become irrelevant.

Once knowledge is encoded, it becomes detached, abstract and is therefore typically expected to be more generally valid than for instance design reasoning, which is encoded to a lesser degree. This gives rise to the illusion of objectivity. On several occasions in the toolmaking study, I encountered collisions between the immediate concrete issues and the subjective reasoning of designing on the one hand and the abstract nature of and objective validity implied in coded knowledge on the other. This was often the case when I envisioned quantitative methods to evaluate outcomes of designing. My attempts at quantitative assessment of visual irregularity (see section 3.8), structural analysis of a new pre-stressed grid type (see section 3.8) and counting different node types contained in grid structures (see section 3.9) all failed, seemingly due to their lack of appreciation for subjective viewpoints or concrete immediacy of given scales of size.
4.2.4 Analogies and distinctions

In the encoding of originally continuous perceptions onto discontinuous representations, two elementary cognitive operations seem to be particularly important. One is the drawing of distinctions as described by Spencer-Brown [1997] and modified by Glanville [2002] and Varela, which can be characterised by the statement “this is different from that”. The other is the drawing of analogies as described by Lakoff and Johnson [1980], which can be characterised by the statement “this is like that”. Similar to the notation used in figure 4.2, figure 4.3 illustrates the role of analogy and distinction drawing when mapping continuous perceptions onto defined states (such as previous or newly constructed knowledge). A mapping between the two lower bars shown in figure 4.2 would deploy analogy and distinction drawing in a similar fashion.

The drawing of distinctions and of analogies are private activities and they cannot be performed for someone else unless in a restrictive fashion. This results in a problem when designing for other designers, as is likely the case in digital design toolmaking for others and as was noted by the critics of the design methods movement (see section 1.2.4).

A typical way of drawing analogies is by using metaphors. Le Corbusier’s description of the functioning of the *Philips Pavilion* as a “stomach” described in section 1.4.3 (see also Evans [1995], pp. 19-37) is a good example. It demonstrates that designing involves the drawing of analogies between things that are yet unrelated to provoke the construction of new knowledge. Other examples include PTW’s foam metaphor for the *Beijing National Swimming Centre* (see sections 1.4.3 and 3.6) or the explanation of design in evolutionary terms given by Frazer [1995] as well as by Dorst and Cross [2001]. I have drawn numerous analogies in the toolmaking study described in chapter 3 such as the one between the constraining
of grid rationalisation and a mechanical digital calculator in section 3.9 and the one between the alignment of faces in grid structures resembling liquid foam and voltage division in electrical engineering (see figure 3.36). Another example is the analogy between procedures for counting different node types in grid structures and spacecraft attitude determination strategies (see also section 3.9). The drawing of the analogy between cellular development and designing was however unsuccessful, the reasons for which I examine below in section 4.2.5.

Figure 4.3: Drawing analogies and drawing distinctions.

An example of distinction drawing is my response to the rationalisation problem posed by the Beijing National Swimming Centre discussed in section 3.9. The clustering of different strut lengths (see figure 3.32) into standardised strut types requires the drawing of distinctions by determining that one strut type is different from another, and if applied in space frame construction, proceeding by manufacturing them in separate batches. One’s freedom in drawing distinctions depends on the tolerances permitted by observer and observed. In space frame construction, this kind of tolerance can be permitted for example by the length-tolerant joint details shown in figure 3.33.

A metaphor is an analogy; a statement that “this is like that” in some specific
way, which is determined by the one who draws the analogy. Someone else may agree in another specific way or disagree. An analogy is thus a way of using existing knowledge to encode new knowledge in somewhat analogous ways. Analogies inspire, by producing variety. But they are, as the term suggests, analogue, not coded and they offer thus no means for controlling. Design tools inspired by nature or biological analogies may encode elements of a toolmaker’s knowledge of a specific aspect of some natural phenomenon or of some biological theory in a new and imaginative way. But they cannot be expected to fully embody the phenomenon referred to in the analogy. The soap bubbles analogy in designing building form can be successful, as the Beijing National Swimming Centre demonstrates. The developmental biology analogy in digital design toolmaking, however, was not successful at all in my toolmaking. Expecting Zellkalkül to design in the way in which Nature generates new and apparently well-adapted forms makes as little sense as hoping that children could chase the bubbles of the Beijing National Swimming Centre through the air.

4.2.5 Pre- and post-rationalisation

The mapping of continuity onto discontinuity can happen in two different ways. One can either know a suitable way of encoding (have suitable previous knowledge) before perceiving a continuous experience, such as an utterance at the other end of a design conversation in which one is involved, or one can have such an experience first and make up a suitable discontinuous encoding system (construct new knowledge) afterwards. These approaches of pre- and post-rationalisation can be observed in some grid structures I developed (see figures 3.22 and 3.20). The time lines of both operations are illustrated in figure 4.4.

During the toolmaking study, I never observed either of these approaches in
4. REFLECTIVE ANALYSIS

Figure 4.4: Pre-rationalisation and post-rationalisation.

their pure form but always mixed in some way. This points towards the idea of co-rationalisation, which suggests the possibility of rationalisation “as-one-goes”. Examining digital (“notational”) and analogue (“non-notational”) representations, Goel [1995], p. 166 refers to an intermediate third form of exchange, which he calls “discursive”. This points to what Glanville [1996] describes as meta-conversation: exchange that aims at making sure similar understandings are constructed on both sides. This type of communication controls the negotiation of knowledge encoding at the border between the analogue and the digital “as-one-goes”.

Trying to support designing with co-rationalising digital design tools that allow the identification of geometric strategies “as one goes”, I merely managed to support a combination of pre- and post-rationalisation, for which the term “embedded rationale” (see section 1.4.4 and appendix K) seems more appropriate. I found no guiding support in the concepts of pre-, co- and post-rationalisation while
designing new types of space grids but I found it easy to apply these concepts to what I did in retrospect, suggesting that applying the concepts of pre- and post-rationalisation is itself a post-rationalisation (see section 3.8).

In this sense, I believe that Woodbury and Burrow’s [2003], p. 517 description of design solutions as points in a design space is a model of post-rationalised experiences of designing. It is thus probably difficult to apply prescriptively. Similarly, my idea to reduce continuous solution spaces to a set of allowed points in a rationalising design tool in analogy to a digital mechanical calculator (see section 3.9 and right of image 3.29) did not acknowledge the nature of designing. Such a tool would encode pre-rationalised knowledge and leave little for the user to choose from or decide. I also believe that Cross’ [1977] distinction between “magic” and “hackwork” can explain what happens in designing in retrospect, that is, post-rational, but that it is difficult to apply a resource that may be used in designing to support “magic” or “hackwork” pre-rationally.

This thesis document demonstrates how the processes of pre- and post-rationalisation apply to any coded communication. The words of the English language I am using here were (pre-rationally) available and associated with what they refer to long before I embarked on this research. What I then experienced as a continuous research activity is now (post-rationally) encoded in the ten stages of the toolmaking study discussed in chapter 3, and in the seven aspects of second-order cybernetics used in this chapter to reflect on the toolmaking study and so forth for the sake of clear communication. In the process through which this document came into existence, pre-rationalisation and post-rationalisation amalgamated into what can be described as co-rationalisation. Herein lies the essential understanding that I gained about the notion of co-rationalising in this study: Pre- and post-rationalisation tend to occur in some form of combination and,
4. REFLECTIVE ANALYSIS

in retrospect and with a focus at broader time frames, may be post-rationalised and identified as co-rationalisation.

Pre- and post-rationalisation can also be applied to qualify methods in science and in design. Philosophers of science distinguish between descriptive methods, which offer explanatory principles for observed phenomena, and normative-prescriptive methods, which offer guidance in the intentional production of some desired condition (see section 1.5.1 and Kantorovich [1993], p. 57). Descriptive methods are encodings that explain phenomena after their observation and are hence post-rational. Normative-prescriptive methods encode knowledge that is to be applied in future action and are hence pre-rational. If designing is itself the process of encoding new knowledge through new ways of encoding knowledge (this is what I am suggesting in this thesis), then the forced introduction and application of previously encoded knowledge (and previous ways of encoding knowledge) in this process results in a collision of interests. This can explain the demise of design methods and of digital tools for designing that are to be applied in contexts other than those of their own origin. It also offers an explanation for the success of some natural or biological analogies for designing and the failure of others. The co-evolutionary model proposed by Dorst and Cross [2001] for the joint changing of the understanding of design problems and of design solutions works well as it simply offers an explanatory and not a prescriptive principle. My cellular development analogy for tool design aimed at prescribing how designing should proceed and it does not work. The cautioning positions reviewed in section 1.3.6 can be assumed to apply primarily to normative-prescriptive methods, which aim to guide designing towards desired outcomes.
4.2.6 Making meaning

Articulations do, in themselves, not contain meaning. It is up to the receiver in a conversation to make meaning from them. Von Foerster [1999] describes as the Hermeneutic Principle that it is the listener, not the speaker, who assigns meaning to an utterance. The listener’s meaning is by necessity not identical the sender’s and only thus can it have value with respect to novelty generation, i.e. to designing. Therefore, there is on the one hand little point in controlling meaning when aiming to generate novelty. On the other hand, this shows the importance of the receiver in conversations. It is due to the receiver’s end that designing produces novelty, encodes knowledge and becomes an epistemological process.

This can explain the freedom designers enjoy to accommodate unexpected observations, in contrast to scientists who, as they were described by Popper [1970] (see also section 1.5.1), follow formal and essentially critical research plans. In developing new irregular grid structures, I assumed that grouping smaller convex polyhedra around a larger one would result in a grid patch that could be repeated in a Cartesian fashion and contain a single strut length and a single node type only. Scientists call such an assumption a hypothesis. Putting this idea into action (using digital three-dimensional modelling tools), I found that the smaller polyhedra do not line up as I initially hoped (see figure 3.22). The stereotypical scientist would at this point, if adhering to a formal, pre-planned research procedure, bring the experimental enquiry to an end and declare the hypothesis as falsified. Seeing how close the result came to a nicely lined-up packing, I did not dismiss the outcome of my design experiment. I saw that tweaking the structure slightly would give a good enough result, at the expense of only one or two new strut lengths and one new node type, which at this point I knew was still a very small set of elements. I accommodated what I observed and accepted it as a possibly good
enough outcome. Scientists with common sense would of course likely do the same or something similar. But the conventional forms of articulation by which scientific knowledge is disseminated tend to exclude the acknowledgement of this role of designing in science. In this particular example of novelty and knowledge generation, the three-dimensional digital modelling tool I used was instrumental. The tool supports, from the toolmaker’s perspective and in Cross’ [1977] words, “hackwork” (see also section 1.2.5) by offering essentially spatial drafting functions. I, the receiver of the tool, made new meaning out of what it showed me in my conversation with it. The sender (that is, the toolmaker) could not have predicted this use in my novelty and knowledge generation.

Observing the use of digital design tools in the postgraduate design workshop discussed in section 3.11, Herr and I intended to follow the formal scientific model of initially laying out what we were looking for in coded, quantitative ways (see questionnaires in appendix H). In the end we did not find much of interest with respect to our previously encoded research plan, and those quantitative findings that indicate unexpected phenomena are difficult to speculate about in terms of possible underlying causalities and implications for future design toolmaking. Other less expected kinds of observations, however, turned out to be of value to us. All cases of tool “abuse” or appropriation in the workshop were, by definition, not anticipated by us, the toolmakers. We nevertheless recognised that these were the uses of our tools that involved novelty generation. The group that showed the chart reproduced in figure 3.44 used it to explain that in their designing they were neither interested in the “low complexity” at the left end of the chart nor in the “high complexity” at the right end but in the kind of medium complexity at the centre of the chart. The group used the tool not only to design but also to better understand what and how they wanted to design (and also used the tool.
to illustrate this use). This suggests that what had been developed as a tool in Glanville’s sense (see section 1.2.6) was, in the same sense, now used as a medium. This indicates again that meaning is made at the receiver’s end of a communication while control is usually in the interest of the sender. It also suggests that the distinction between tools and media is post-rational and descriptive only, similarly to the distinction between “magic” and “hackwork”, and that it is of little help as a prescriptive guiding principle in digital toolmaking (see section 4.2.5).

Other observations made during the toolmaking study that exemplify how the making of meaning and the generation of novelty and knowledge depend on the receiving end of design conversations include users’ requests for design tool modifications and extensions in the case of Zellkalkül (see section 3.3), the Tsunami Memorial Design Competition (see section 3.10) and in the postgraduate design workshop (see section 3.11). In these cases, users received the tools as expressions in design conversations with me, the toolmaker. As the tools and the operations they offered did not match the users’ understanding of how their designing should be supported, they made new knowledge which I was then asked to accommodate by changing the tools.

4.2.7 Supporting knowledge generation

So far, I have established that novelty and knowledge generation is a co-rationalising activity that depends on the (receiving) designer’s ability to accommodate and encode unencoded perceptions within design conversations. As implied in the term co-rationalising, this activity also involves the use of already existing knowledge that is available to the designer. This can be prior knowledge of the designer or it can be knowledge that is encoded in resources the designer uses, such as tools. Digital design tools, by definition implemented in software, quite obviously
contain encoded knowledge. In the self-referential replication of tools in which tools are designed by way of using tools and are themselves used in designing yet other tools, the introduction of previous knowledge into designing seems barely avoidable. What seems crucial, though, is the designer’s freedom in choosing the resources deployed in designing and their manner of application. In the postgraduate design workshop the members of one group used one of our digital design tools to think about the level of geometrical complexity they wanted to aim for and, on top of that, proceeded by using the tool to illustrate this strategy in a self-reflecting presentation. This use was entirely unanticipated by the toolmaker. I assume that in that moment, at least one of the group members had made new knowledge, thereby transforming him- or herself, being in a conversation (system) with the tool, looking inwards. What happened in this conversation is inaccessible to me (see section 4.2.2). But, in my view, judging by the group’s articulations in both self-reflecting presentation as well as in design outcomes, this moment constituted one of the best contributions of novelty and knowledge of the workshop. In order to support moments like this, there would be no point at all in the toolmaker insisting that tools be used as intended. On the contrary, the toolmaker must relinquish control, which may appear counter-intuitive and illogical to enthusiastic aspiring digital design supporters, as I experienced in my toolmaking study. It is however not illogical, and this reflective analysis and the second-order cybernetic understanding of designing explain why. It is the responsibility of the receivers in design conversations to make meaning. It is thus the responsibility of supporting toolmakers not to encode knowledge (that is, not to exercise control by drawing distinctions and analogies) on behalf of the receivers, since the result would be restricting. Restriction of communication tends to be practiced by senders who do not understand or who are not interested in novelty
(and knowledge) generation. In designing, it offers very limited support.

### 4.3 Primary findings

With respect to the primary research questions put forward at the end of the review of literature and work of others in chapter 1, the following primary findings and explanations can be obtained from the toolmaking study in chapter 3 and the reflective analysis in chapter 4:

1. Can natural or biological analogies guide the development of digital design tools?
   
   As noted in section 3.12, and as far as observations made during the toolmaking study are concerned, the answer to this question is negative in cases where principles explaining natural phenomena are followed strictly, aiming to reproduce these phenomena (as in the case of cellular development and *Zellkalkül*).

   This primary research question 1 can be answered positively where the notion of “guiding” is used more loosely and observations in Nature are employed as informal inspirations (such as using soap bubbles as inspiration for building form).

2. Can the development of digital design tools be justified after the abandonment of design methods?

   As noted in section 3.12, and as far as observations made during the toolmaking study are concerned, the answer to this question is mostly negative with exceptions where tools are developed and applied within the same context. The reflective analysis outlined in this chapter offers the following explanation for this answer: The drawing distinctions and analogies, and thus the generation of knowledge and novelty are not possible on behalf of others. Digital design tools (being “solidified” aspects of design methods) and design methods represent
encoded knowledge, which, when applied in designing, results in the restriction of the designer’s making of new meaning. Exceptions are those cases in which digital design tools (and methods) are made within the context of applied designing, where they result from an internal conversation with the designer and adapt according to an outside conversation in which the changing understanding of where the design process is negotiated.

3. Can digital design tools be generalised from the contexts of their production to other contexts and to other users in the way non-design tools like scissors and bulldozers can?

As noted in section 3.12, and as far as observations made during the toolmaking study are concerned, the answer to this question is negative. The reflective analysis outlined in this chapter offers the following explanation for this answer: Designing is concrete while generalised tools or methods are abstract. It is the designer who must subjectively draw distinctions and analogies to encode new knowledge. This responsibility cannot be performed on behalf of others (see also the explanation for primary research question 2).

4. Can the concepts of pre-, co- and post-rationalisation offer guiding principles for the development of digital design tools?

As noted in section 3.12, and as far as observations made during the toolmaking study are concerned, the answer to this question is mostly negative in the case of pre- and post-rationalisation while the notion of co-rationalisation has apparently guided digital design toolmaking with a successful outcome. The reflective analysis outlined in this chapter suggests that the answer to this question is entirely negative and offers the following explanation for this answer: Co-rationalisation as a process that utilises existing coded knowledge and uncoded knowledge to generate new coded knowledge applies to any designing.
Pre- and post-rationalisation seem to be reduced idealisations which rarely occur in their pure forms in designing. For these reasons, the concepts of pre-, co- and post-rationalisation all seem to be of only descriptive value and have failed in offering prescriptive guidance in designing grid structures or in determining operations that are to be supported by digital design tools.

4.4 A descriptive model of designing

As noted in the review of literature and work of others on tool use in animals in section 1.2.1, earlier research regarded tools as objects that fulfil certain criteria, that are applied in linearly directed ways to achieve intended effects and that are evaluated in terms of measurable efficiency. As was further noted, more recent research is concerned more with the interaction between user and tool, and attributes characteristics of tool use (and the very functioning of objects as tools) to observers of such loops of interaction. A focus on linear control has been replaced by a focus on meaning constructed by observers.

By the same token, it is interesting to relate for example “A pattern language” by Alexander et al. [1977] to Dorst’s [2003] “Understanding design”. Both books are essentially collections of small, easy-to-read chapters, each of which lays out some previously gained design knowledge with the essential objective of making this knowledge re-usable by their respective readers. Both books present designers with ideas concerning design and, in my reading, both invite their readers to appropriate the various presented ideas without seeming to insist on the ideas’ necessary validity beyond their authors’ personal views. While Alexander’s pattern language has been widely rejected in the design field, Dorst’s “Understanding design” was published 25 years later, so far without provoking a broad rejection. I believe the key difference between the two books lies in the nature of the directedness
4. REFLECTIVE ANALYSIS

of the ideas they present. “A pattern language” offers (linearly directed) guidance regarding the way design outcomes should be structured while “Understanding design” offers, as stated in its subtitle, (circular) “reflections on being a designer”. Again, a focus on linearly directed prescription has been superseded by a focus on self-transformation. The difference is that between first-order and second-order cybernetics and between the statements “A has a controlling effect on B” and “A has a transforming effect on itself”. The prior linear cause-and-effect model is successfully applied in many contexts. Examples include the grinding of a blade to sharpen it, the boiling of water to clean it, the pushing of a door to close it, the ignition of fuel to harness its energy and the toggling of a switch to turn the light on or off. With its emphasis of control, however, this model precludes designing. The latter, self-transformational model appears to describe designing itself. As Glanville [1997] points out, the relationship between the two is the relationship between a track and the wheel that leaves it. I began to understand Glanville’s wheel analogy after I imagined novelty generation as a track-laying train such as the one shown in figure 4.5 (I discuss this analogy in more detail below). Interestingly, the Jones quote on design methods cited in section 1.2.4 also refers to tracks and, similarly to Glanville’s notion of restriction, Jones points out that the tracks in question offer choices that are not free. According to the understanding of tools as “solidified” aspects of methods, the word “methodology” in the Jones quote can be exchanged with “design tool” to the effect that, in accordance with the observations I made during the toolmaking study presented in chapter 3, the resulting variation keeps all its validity (at least in my view):

“[Design tools] should not be a fixed track to a fixed destination, but a conversation about everything that could be made to happen. The language of the conversation must bridge the logical gap between past and future, but in doing so
it should not limit the variety of possible futures that are discussed nor should it force the choice of a future that is unfree.” (based on Jones [1992], p. 73) In other words: Making digital design tools for others is like laying train tracks for travelers who must drift to unknown destinations.

Following the observations reported and reflected in this thesis, I developed a descriptive model, which I find useful in explaining to myself processes of novelty and knowledge generation in design and science in analogy to a moving, track-laying train. This model integrates Reichenbach’s (see Salmon [1970]) notions of contexts of discovery and of justification, Wilden’s [1980] and Goel’s [1995] distinction between the coded and the uncoded (or the analogue or non-notational and the digital or the notational) as well as Cross’ [1977] distinction between “magic” and “hackwork”. It also positions the domain of relativist thinking in relation to the domain of positivist thinking as suggested by Dorst [2004]. The model furthermore offers explanations for the distinction between discovery and invention as discussed by Foerster [2003] and for the notion of co-rationalisation suggested by Aish (see section 1.4.4). The model furthermore accommodates the second-order cybernetic understanding of designing as a conversation, presenting my concept of design conversation as a track-laying train.

In this descriptive model, illustrated in figure 4.5, novelty generation is an active process, represented by the moving train. Along with the train move two lines, one of which separates what is yet unknown from what is known and the 

---

3In figure 4.5 the tracks laid by the train are visualised in a straight line. The process that precedes the laying of the tracks is most likely not straight at all, possibly involving curves, turns, junctions, dead-ends and the like. Once tracks of the described kind have been laid out they appear straight and directed when looking back at them, suggesting track-laying processes to be simpler and more straightforward than they usually are.
Figure 4.5: Descriptive model of novelty-generating processes in design and science in analogy to a track-laying train.
other one of which separates what is yet uncoded from what is encoded. The train heads for the context of discovery, which is the unknown. This happens all the time, whether expected or not, intentional or not, with a purpose or without. Having an open mind while talking to another person is one way to enter unknown territory. Confronted with the unknown, we begin establishing a familiarity and possibly an intuitive understanding of it. Knowledge formed at this point is not yet structured by logic, language or the like. It is uncoded. If one wishes to go further, then one must engage with this knowledge in what Glanville [1999] describes as a conversation. The uncoded (or analogue) knowledge is at this stage rich and authentic but it is also ambiguous and vague. It can be described as “states” at the opposite end of the conversation for which we do not (yet) have our own corresponding “states” or coded concepts. There are now three options to deal with the uncoded knowledge. The first option is to do nothing that would impose coding structures onto this analogue knowledge. Analysis, the introduction of structure and of representations such as words can be harmful or simply not do justice to some things we can feel or know in analogue ways.

The second option is to restrictively “pigeon-hole” or to “shoe-horn” the analogue knowledge into the structure of the coded knowledge that one already possesses. In this case there is no novelty-generation because there is no encoding of new knowledge, just use of existing knowledge. What was previously unknown is now described in terms of what one already knew previously. The third option is to allow the uncoded knowledge to inspire new “states” and new knowledge in one’s

---

4On display at the Altes Museum in Berlin is the notebook of the egyptologist and architect Ludwig Borchardt who found the famous Bust of Nefertiti in Egypt in 1912. After finding the statue he writes: “Description is useless. It must be seen.” (“Beschreiben nützt nichts, ansehen.”)
4. REFLECTIVE ANALYSIS

thinking. In this case the uncoded knowledge changes one and one learns. This involves the use of prior knowledge but more importantly it also produces new knowledge. Producing new “states”, one encodes what was previously uncoded. In this sense, perception must precede abstraction and the qualitative must precede the quantitative. In this translation process, one draws analogies and distinctions in a private and personal way. For this process, the term “magic” seems very appropriate (see section 1.2.5 and Cross [1977], pp. ). In its encoded form, this knowledge is now symbolically represented and it can as such be externalised and shared. The process of externalisation is the process of laying tracks. Examples include scientific writing and design competition entries. Just as sleepers of a railway track can be individually referenced and distances between them counted, compared and so forth, it is possible to relate coded knowledge to other coded knowledge, to compare it and to assess it formally. The process of laying tracks is what Cross (ibid., pp. 141 ff.) refers to as “hackwork”. It involves the application of previously coded knowledge (tracks), possibly in the form of (prescriptive) methods or tools. Similar to actual tracks for actual trains, the tracks laid out by knowledge-encoding processes restrict deviation from defined paths (reduce variety) and allow smooth arrival at intended goals (control).

Using encoded knowledge to encode previously uncoded knowledge, this process always integrates processes of pre- and post-rationalisation. Designing (including novelty generation in science) is hence always a co-rationalising activity. Once the tracks are laid out into the context of justification behind the train, they are available to others for scrutiny and criticism as well as for application and appropriation. Scrutiny, criticism and application in scientific contexts of justification differ from their counterparts in designerly contexts of justification. In designerly contexts of justification, criticism is not required to adhere to coded
and formally scrutinised knowledge. It can be personal, intuitive, emotional and even irrational. In scientific contexts of justification, criticism is required to use other coded and formally scrutinised knowledge as a frame of reference. Based on the used frames of reference, both the encoded knowledge (the tracks left by the train) and criticism of it can be deemed “valid”. With every process of formal scrutiny depending on another process of formal scrutiny, scientific criticism involves a recursive framework that ultimately depends on postulates that are no less questionable than the validity of knowledge encoded in design contexts (rather, they are knowledge that was encoded previously in design contexts). Engaging with what lies ahead of the train requires a relativist mind set while that, which is left by the train facilitates a positivist mind set (see Dorst [2004]).

The distinction between finding and making (as raised in chapter 0) is the distinction discussed by Foerster [2003], p. 285 (see also section 1.5.4) between discovery and invention. Finding (discovery) happens at the line between the unknown and the known (see figure 4.5) every time something unknown is encountered, perceived and maybe comprehended at a non-linguistic, non-logical level. Making (invention) takes place at the line between the uncoded and the coded (see figure 4.5) every time symbolic representations are negotiated (by drawing distinctions and analogies internally and by using meta-conversations externally) and assigned to something that was previously comprehended at a non-linguistic, non-logical level only. The term “form finding”, used by Frei Otto [2001] (see section 1.4.1) and by myself in the earlier phases of the toolmaking study discussed in chapter 3, can thus be described as a misnomer. Encoding new knowledge in his work, Frei Otto (see for example Otto [1987]), is a form maker much more than he is a form finder.
4. REFLECTIVE ANALYSIS

4.5 Discussion

In the introduction of this thesis (see chapter 0), I contextualised the work presented in this thesis with the question of what it is that defines us as humans. Creative inventiveness, the production and use of language as well as the production and use of tools are frequently cited as possible criteria that distinguish us from other animals. This suggests that what distinguishes us as humans happens in Glanville’s wheel (or my train) much more than on the tracks left by it. The Pythagorean view underlying much scientific research until today, expressed in Galileo’s dictum that “the book of nature is written in the language of mathematics” (see section 1.5.1 and Kantorovich [1993], p. 59) implies that any understanding of the world must be limited to what can be formally encoded, since everything there is in Nature is assumed to be coded a priori. This view runs the risk of ignoring all that which is relevant because, and possibly just for as long as it is uncoded. I believe it is safe to assume that this danger has been and continues to be amplified by the role of digital computation in contemporary science (and analoguously, in our discipline, by formal and quantitatively oriented attempts at research for and about design). It can thus be argued that “objective” science and digital technology are potentially at odds with that which defines us as humans. I believe to recognise Pythagorean and positivist views in much of the education that I received prior to this research\(^5\) and I suspect it may be blamed in part for my positivist approach to designing and design support at the beginning of the work outlined in this thesis. As Wilden (see appendix M) notes, tertiary education finds comfort in coded knowledge and avoids uncoded, analogue knowledge. Developing a sensitivity

\(^5\)My undergraduate training as a teacher was largely centred on theories concerned with methods, formal analysis, prescriptive planning and orientation towards fixed goals – rocket science, essentially (see sections 0.2 and 1.2.4).
for processes of knowledge and novelty generation is not a simple task, and it is impeded by an education that focuses on coded knowledge and linearly directed control. I am afraid with this educational focus damage is done every day.

In chapter 0, I likened the practice of language use to supporting the practice of language use, noting that using a language is probably easier than writing a book on its grammar. The reason is that the latter self-referentially involves using language. I chose language to make this point as its use has been suggested to be one of the characteristics that sets us apart from other animals. I went on to suggest that studying a microscope with the only observational method available involving the same microscope itself would be more challenging than say studying one microscope through another one. The challenge of self-reference in observation has been noted by Foerster [2003], p. 285, who reminds his readers that the observer is not independent of her or his observations. This applied to myself observing my research through design in the work presented in this thesis. The relationship between everyday use of language and the more demanding writing on grammar now turns out not to be the best analogy to the relationship between designing and the support of designing. Everyday language use happens behind the train referred to in my model (see figure 4.5). It generally uses existing tracks and, accordingly, happens fairly automatically. Designing, being a novelty and knowledge generating activity, involves an engagement with the unknown, in which personal analogies and distinctions must be drawn. In my model, it is presented as the laying of tracks. This can, it seems to me, hardly happen automatically. It involves the constant making of new knowledge, which by definition is not known beforehand. This has consequences on the possibility of supporting designing by means that require advance coding of procedures that would force already existing knowledge onto the process of designing. The
challenge of supporting designing thus does not so much lie in its level of abstraction, but more in its making of new knowledge.

There seem to be two options in which designing can still be supported by making digital tools. One is not to encode knowledge in digital tools in advance but in parallel to, and involved in the supported design effort (this approach is applied successfully in advanced architectural practices). The other is to avoid control and restriction of novelty and knowledge generation and to focus on re-presenting what is exchanged in design conversations in ambiguous ways that stimulate the making of new meaning  

6 (this approach is what Glanville describes as medium use. See section 1.2.6 and Glanville [1992], [1994a]).

Also in chapter 0, I raised the question of whether the first rudimentary stone tool was found or made. The track-laying process of designing illustrated in figure 4.5 is an internal, cognitive process. Assuming this model captures the nature of designing (track laying) as opposed to other practices (track using), I would now argue that it does not matter so much whether the first stone tool was a usefully shaped physical object found in the external environment or whether it was an inconspicuous physical object that was intentionally shaped to afford useful properties. This is merely the difference between post- and pre-rationalisation, which, by analogy to my toolmaking experiences, can be assumed to have occurred in some combination. In both scenarios the challenge was more one of an internal making process; a process of drawing the analogies and distinctions that led to seeing new possibilities  

7.

---

6 The first option is, in reference to Cross [1977], more “hackwork centred” while the second one is more “magic centred”. Both options have also been discussed at the end of section 3.11.

7 Designing can thus be said to be less about “what the hand does” but more about “what the eye and the mind see”. In the context of scientific knowledge production,
4.6 Summary and conclusions

Designing, in the sense of generating novelty and knowledge, can be described as a conversational activity with conversation partners of various kinds including others, imagined others, interactive software and things found in the physical environment. This activity seems to be essentially the same in the practices of design and science. It engages with what is unknown and uncoded and encodes it in expressions that can be shared, commented on and applied in constructing yet other knowledge in other design conversations. At the heart of this lies the designer’s open-minded readiness to recognise and accommodate that which is not immediately understood by drawing new analogies and distinctions. This production of new knowledge implies the designer’s self-transformation. The production of knowledge depends on rationalising operations by which existing coded knowledge and as yet uncoded knowledge are transformed into new coded knowledge. Pre-rational and post-rational thinking amalgamate at different scales, and in ways that are both difficult to predict prospectively and to re-trace retrospectively, effectively rendering any designing a co-rationalising activity. In this activity, restricting control, which can result from the forced introduction of existing coded knowledge, for example in the form of digital tools or prescriptive methods, should be relinquished. Providing support for this personal activity is not a matter of offering generic tools that can be linearly directed at known goals.

biomedical scientist Hwang Woo-suk was recently found to have fraudulently reported a pioneering success in cloning human embryonic stem cells, arousing indignation in and beyond the scientific community. Later it was found that in their overambitious work Hwang and his colleagues succeeded in extracting cells from eggs that had undergone parthenogenesis, which is also considered a pioneering breakthrough (see Kim et al. [2007]), but which the team failed to notice.
in the way non-design tools such as screwdrivers and bulldozers can. Providing support for this activity in the form of digital tools is a matter of enabling a process of self-transformation based on the generation of new knowledge. Observations made in this thesis suggest that there are two possible strategies available to the digital design toolmaker for achieving this:

1. The operations supported by the tool are not aimed at expressing design outcomes but at the user’s engaging in a variety-generating conversation in which the unknown is encountered and to be accommodated by newly constructed knowledge. If this approach is followed, the precision and control offered by digital tools are likely disadvantageous and limiting. Toolmaking following this ("magic-centred") approach can amplify its potential to benefit others in designing if it supports rich variety as is commonly found in analogue systems.

2. The toolmaker is “on board” with the design team in the context in which the tool is applied. The toolmaker modifies the tools according to the changing understanding of the design challenge and along with the changing design objectives that are being determined through the design process. If this ("hackwork centred") approach is followed to express design outcomes, the precision and control offered by digital tools can be advantageous at stages during which ideas are externalised physically.

4.7 Contributions

As stated above in section 4.5, this thesis is less about processes of the kind “A has a controlling effect on B” and more about processes of the kind “A has a transforming effect on itself”. In this sense, probably the most significant contributions of the work presented here have offered themselves to me “in the wheel” of doing research though designing, as a way of learning about designing by designing (tools (for
designing (tools (for ...))). This learning can essentially not be shared with others (since it is mine and past). This thesis document is the track left by that wheel. The track can be shared, the wheel cannot (see also section 0.3).

Part of this track are my descriptions of grid topologies, rationalisation strategies and the geometrical data I present in this thesis, including the polyhedra geometry and packing analyses listed in appendix E, which may save others time and effort of searching or constructing this data from scratch. I also showed that, valuable as geometry rationalisation for visually irregular architecture may be, the capacity to perform it is not of critical importance at conceptual design stages. This may inform others in their strategic decision-making when proposing design ideas. The above points are however not amongst those contributions resulting from this thesis that I think could be of most value.

I am offering a first-person account of a digital design toolmaking study, including some of its successes and some of its failures and dead-ends, which I then reflect within a single theoretical framework. Accounts of this kind seem to be rare and overshadowed by the great number of post-rationalised, outcome-focused reports on digital design toolmaking. I examined the possibilities of informing those who make digital tools for designing in their efforts by reflecting on my toolmaking study within a theoretical framework of second-order cybernetics. This theory, on the one hand, offers explanations for much of what I experienced in making digital tools for designing. The experiences made during the toolmaking study, on the other hand, exemplify much of second-order cybernetics, which has so far been largely put forward in abstract terms. In this sense, I believe my toolmaking work and second-order cybernetics contribute to each other. Readers from the area of geometry and digital design toolmaking may gain useful insights into second-order cybernetics while those with a background in second-order cybernetics...
cybernetics may gain insights into this theory’s implication for designing based on
the practical experiences I refer to. From this, and from what others have published
previously, I derived a personal descriptive model of knowledge generation in
design and science. In doing so, I identified parallels between design research
theory and second-order cybernetic theory of positions and views observers can
take in relation to systems. With this thesis, I make this model and the identified
parallels between different theories available to the field, to be scrutinised and
possibly adopted, applied or extended by others.

This model, the theories it accommodates and the theories upon which it
is based, as well as my account of my own self-transformation through this
research, aim to contribute an implicit critique of knowledge generating practices
in contexts of designing, science and education, which assume the existence of an
objective reality, prioritise methodical processes, formal specifications, rational and
realist reasoning over subjective and relativist viewpoints, differences in individual
intuition, and the values of ambiguity and uncertainty.

4.8 Future work

With this thesis, a new section has been laid out and added to the existing network
of tracks that encode what is known about designing. The track-laying does not
stop at the end of this thesis as the designing train goes on. As a next step, others
may take a look at what has been written here, make their own meaning, construct
their understanding from it and see how it can relate to their knowledge and
experiences. Subjecting this thesis to scrutiny, modification, extension or rejection
will help developing knowledge on designing and the possibilities and limits of
supporting designing.

Opportunities for further research and future work based on this thesis extend
across the areas of design research, design education, design practice and design theory. My suggestions regarding possible future applications of outcomes of this research are, in accord with the arguments put forward in this thesis, not to be understood as an “action list” aimed at instructing or controlling others. Those who are addressed (including myself) shall make their own understanding of what I suggest and their own decisions when building future work upon the work presented in this thesis. Those who make, apply and evaluate digital tools in design research contexts and in design education may investigate new strategies to direct their efforts to applied design projects. Digital design toolmaking as well as research into digital design toolmaking do, as it seems, benefit from practical involvement. Future research will have to examine if and how far specific observations presented in this chapter, my personal explanatory model for my observations as well as my conclusions may be generalisable beyond the scope of this thesis. The approach of developing tools that are made specifically to engage in novelty and knowledge generating conversations, independently from expressions of physical design outcomes, is not yet well understood. Examples of such tools and experiences with their applications are rare. This research direction appears promising and it should, in my opinion, be investigated. Designers and design supporters (educators, toolmakers and so forth) who emphasise the validity of objectivist and realist reasoning, formal and prescriptive procedures and control in their work, in their products, methods, tools or in their theories may consider opening their minds towards subjective and relativist reasoning, the informal and the values offered by that which is not immediately understood. Practitioners, educators and theoreticians of science who ascribe the innovative capacities of scientific research to serendipity and surprise only may benefit from familiarising themselves with the particular forms of thinking and educating by which the
design field nurtures novelty and knowledge generation.

I regard my self-transformation (gain in knowledge) reported in this thesis as a valuable experience. It has equipped me with a stronger vocabulary to think and to talk about designing and about supporting designing on the one hand and with a better sensitivity towards designing and towards listening in design conversations. Without the help of others, my self-transformation would not have been possible. Similarly, I hope that in my future work I can help others in transforming themselves and in making their own approaches to designing that acknowledge both how human this activity is and how fragile it is when subjected to control by that which is already known. I hope this thesis will be a beginning for my own further self-transformation through further exploration of the relationship between second-order cybernetics and designing.


Aicher, O. [1991], Analogous and Digital, Ernst & Sohn, Berlin.


Alcock, J. [1981], ‘Animal tool behaviour. the use and manufacture of tools by animals by benjamin b. beck (review)’, The Quarterly Review of Biology 56(2), 231.


5. REFERENCES

AIA/ACADIA Fabrication Conference, Toronto, Ontario, University of Waterloo School of Architecture Press, pp. 325-335.


Baggott, J. [1990], 'Serendipity and scientific progress', New Scientist 125, 67–68.


Barrow, L. R. [2001], 'Evolving communication', Architecture Weekly 59, T3.1.


Broadbent, G. [1979], ‘The development of design methods’, Design Methods and Theories 12(1), 41–45.


Burry, M. [2003], ‘After Gaudí - innovation through appropriation’, Inaugural lecture at RMIT University.


5. REFERENCES


Cross, N., ed. [1984], *Developments in Design Methodology*, Wiley & Sons, Chichester.


Foerster, H. [1999], 2 x 2 = grün, supposé, Köln.

Foerster, H. [2003], Understanding Understanding, Springer Verlag, New York, Berlin and Heidelberg.
5. REFERENCES


Frazer, J., Tang, M.-X. & Jian, S. [1999], ‘Towards a generative system for intelligent design support’.


Fuller, R. B. [1975], Synergetics, Macmillan, New York.


Gamma, E., Helm, R., Johnson, R. & Vlissides, J., eds [1995], Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, Reading.


Glanville, R. [2003], ‘Machines of wonder and elephants that float through air’, Cybernetics and Human Knowing 10(3-4), 91–106.


Hansell, M. H. [1987], ‘What’s so special about using tools?’, New Scientist 1542, 54–56.


Kieser, D. G. [1815], Phytotomie, oder Grundzüge der Anatomie der Pflanzen, Cröcker, Jena.


Leach, N., Turnbull, D. & Williams, C., eds [2004], Digital Tectonics, Wiley-Academy, West Sussex.

Lecky-Thompson, G. W., ed. [2005], Corporate Software Project Management, Charles River Media, Hingham.


Makowski, Z. S. [1963], Steel Space Structures, Düsseldorf, Verlag Stahleisen.


Medawar, P. B. [1979], Advice to a Young Scientist, BasicBooks, New York.
5. REFERENCES


Munro, D. [2004], ‘Swiss Re’s building’, Nyheter om Stålbyggnad (3), 36–43.


Otto, F., ed. [1972], IL4 Biology and Building, Institute of Lightweight Structures, Stuttgart.


Raymond, E. S., ed. [1999], *The Cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary*, O’Reilly, Sebastopol.


5. REFERENCES


Thomson, W. [1887], ‘On the division of space with minimal partitional area’, *Philosophical Magazine* 24(151), 503.


Weaving, A. [2004], High-Rise Living, Gibbs Smith, Layton.


Designing (tools (for designing (tools (for ...))))

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

Thomas Fischer
Dr. phil. (University of Kassel)

Volume II
Appendices and Glossary

School of Architecture and Design
Faculty of the Constructed Environment
RMIT University
February 2008
“A.1 Esherick [1963]: Problems of the Design of a Design System

Are quantitative approaches to design likely to be restrictive or limiting?
Are they likely to be authoritarian or dictatorial?
Are they likely to produce a monolithic social order or one so strongly structured and framed that there is no freedom of movement or choice?
Are quantitative approaches to design likely to generate systems which will ultimately become closed systems in spite of every effort to keep them open, forcing upon us a formalistic design process insensitive to the demands of a living and growing human system and leading to a fixed and rigid environment?

A.2 Cross [1977]: CAAD system checklist

System design
How will the particular computer system be chosen?
  What criteria will be used to select a system?
  Who will be responsible for deciding on the particular system?
  Who will be consulted?
Who are the system designers?
  What is their experience and background?
What are the objectives of the system?
APPENDIX A - “LAUNDRY LISTS” FOR DESIGNING DIGITAL DESIGN TOOLS

What is it meant to achieve?
Why is it being implemented?

What functions will the system perform?
  Does it replace existing functions of the design office?
  Does it change existing functions?
  Does it ignore some existing functions? If so, what will happen to those functions?

How will system functions be selected?
  What programs or other aids will be incorporated?
  Who will decide on system functions? With what criteria?
  How will functions be updated or otherwise modified in the light of experience and further developments?

Human factors
What ergonomic or human factors attention has been paid to the design of the ‘interfaces’ between computer or architect?
  Are the input data relatively easy to prepare and present to the computer?
  What forms does the output take? Is it readily understandable?
  Does the output genuinely aid decisionmaking in design?

Personnel
Who will immediately be affected by the implementation of the system, and in what ways?
  Have these people been consulted? What is their reaction?
Who will operate the system?
  What training will they need?
  What work pattern will they be required to adopt?
Who will be barred from operating the system?
  Will this be because of lack of training, or by explicit ruling, or by system design and accessibility?
Who will be responsible for the ‘machine minding’ tasks?
  Who will punch cards and tapes?
  Who will write programs?
  Who will deal with errors and breakdowns?

Efficiency
What is the expected increase in efficiency of the design process – that is, what reduction in man-hours is predicted?
  Has the predicted time saving been tested?
  Where do the savings occur?
  What other savings have been assumed in order to make the system justifiable and viable?

Employment
What effect is the computer system expected to have on employment?
  Will there be a reduced need for design staff?
  Who will be affected?
Office organisation
How will the design office be restructured to incorporate the computer system?
  What will be a normal project team?
  How will it be organized?
  How will teams change from project to project?
How will the management structure of the office change and adapt?
  Will there be changes in the responsibilities held by the members of the office?
  Will the management ‘pyramid’ be affected?
  Will promotion prospects be affected?

Building team organisation
How will the work of other members of the building team – quantity surveyors, engineers, etc – be affected by the computer system?
  Who will have access to which files?
  How will information swapping between files be coordinated?
  What will be the team structure with the new system?

Working pressure
How much of the designers working day will be spent working on-line with the computer?
  Are on-line periods of intensive work fairly brief and separated by longer periods of a more relaxed pace?
  What other steps will be taken to reduce the expected stress of computer-paced work?
Who will have access to records of the designer’s work with the computer?
  Will there be someone ‘looking over the designer’s shoulder’, checking his decisionmaking records?
  What will be the filing system for keeping records?

Design process
What structure for the design process does the computer system assume and impose?
  On what model of the design activity is the based?
  What structure to design problems does it assume?
  What approach to design and problemsolving does it require its user to adopt?

Design solutions
What constraints will the computer system impose on the kinds of buildings that can be designed?
  Will it accept a particular building system, or a few systems, only?
  What limitations are there on the shapes, forms and arrangements that it can handle?

Relations with clients and other users
How will the computer system affect the designer’s relationship with the project client and the building users?
Will the client and/or users have access to the computer data and outputs? 
Will they be able to participate in the computer-aided design process? 
Will the computer system tend to ‘democratize’ or to ‘bureaucratize’ the design process?

A.3 Glanville [1992]: How to treat computing as a medium

There are two answers that I [Glanville] know of. They operate at different levels. The first is the harder. It is to LISTEN. To what the computer is telling, is offering – if I may be allowed a personification that is not intended to be animistic.

By this, I mean not to tell the computer what I want, not to order, but to listen, to watch, to leave myself open to whatever it offers (without censorship or even evaluation in the first instance), letting it take part. I do not know what this is. I repeat, I DO NOT KNOW WHAT THIS IS: this is axiomatic.

What this is like is tutoring architecture. In my world, the teacher’s job is to listen to the student in order to find out what he is trying to do, and to help him do this (or decide not to). This involves the assumption of ignorance – a conceptual carte blanche and a willingness not to know better – the abdication of expertise. That is why I say I must know even less than my students to teach them this, which I refer to as “teaching from behind”. It’s a very hard trick, one of which I am by temperament and skill extra-ordinarily ill-fitted, and is in complete contradistinction to that other soil of architectural teaching, in which we lead, we demonstrate our expertise and we instruct.

It means dropping preconceptions and pre-conditions, and leaving the toolkit behind. It means not knowing what the use will be, being surprisable and alert and open, and, quite possibly, ending up a blind alley – or at least not being an architect any more! It means remembering that for the first time inventor, the wheel is NOT being re-invented. It means sharing the novelty and the bemusement and the surprise and the thrill and the delight. And the hurt. It is very difficult, and I think it involves a very great deal of courage. And innocence.

The second one is more straightforward: it is not an attitude but ways of doing this (could these be tools?). I can list some: I am sure everyone else could add as many. I wish you would.

recognising and grabbing hold of serendipity 
play
association
brainstorming
shating (respect, not ownership, of ideas)
abuse (use in the wrong, unintended way, or beyond the intended range)
distortion
collage and mosaic
following rules to the bitter end
interaction
the effect of another on “my” randomness, change, automatism, accident accepting the system’s performance as it is (inter)active tagging within an environment of discipline, criticism, honesty, rigour and quality.
APPENDIX B

ZELLKALKÜL’S DEVELOPMENTAL SCRIPT FUNCTIONS

The language in which the rules for each cellular identity’s behaviour are expressed is an ECMAScript (see McComb [1998]) dialect. This is a C-like, higher-level interpreted language and a relative to scripting languages supported in Web browsers such as JavaScript™. It is object-oriented and offers the typical features known from other high-level languages such as assignments, variables of different types, loops, conditionals, definition of user-defined objects and access to libraries, for example of mathematical functions. The ECMAScript interpreter has been extended to include the following scripting functions to allow the user general control of the cell objects.

```plaintext
getID();    // returns cell ID
getID(x,y,z); // returns ID of cell at position (x,y,z)
getCode();  // returns cell code as string
getCode(x,y,z); // returns cell code of cell at position (x,y,z) as string
getCoordinate(); // returns cell’s x, y and z coordinates
```
createCell(x, y, z, ID, f); // creates new cell with identity ID at (x,y,z), f=force
setCode(ID,string); // replaces code of cells with identity ID with string
copyCodeTo(ID); // replaces code of cells with identity ID with own cell code
setActive(bool); // includes/excludes cell in/from execution cycle
getCellCount(); // returns current total number of cells
wait(time); // sleep for time ms (useful to allow screen update)
setState(i); // set own automaton-state (int number)
getState(); // returns own automaton-state

Additionally, the scripting language was extended by the following six functions, which are modelled to imitate activities of natural cellular development.

split(n, m, f); // split into two cells of identities n and m, f=force
move(d, f); // move to direction d
differentiate(n); // change own identity to n
differentiate(x, y, z, n); // change ID of cell at position (x,y,z) to n
kill(x, y, z); // kill (delete) cell at position (x,y,z)
die(); // commit suicide

The argument f (as in force) used in some of the functions determines how the system deals with a cell that already exists at a position that is to be occupied by calling the respective function. The three options are:

0: not to create the new cell and allow the existing cell to remain at its position
1: creating the new cell and pushing the existing cell out of the way into a direction of little resistance
2: deleting the existing cell and placing the new one into its position

If called as shown above, the functions will be effective for the currently executed cell itself. It is possible to perform the functions on neighbours of the calling cell as shown below, whereas i denotes the direction of the called neighbour according to the face reference numbers 0 to 11 similar to the face numbers 1 to 12 shown in figure E.3.

neighbour(i).split(n, m, f); // instructs neighbour cell in direction i to split.
With the implementation of the parametric surface geometry control discussed in section 3.5 the following two script functions were added to set and to read the "virtual pressure" of individual vertices of cells.

```plaintext
setTension(v, p); // set pressure p at vertex v
getTension(v);    // read current pressure at vertex v
```
This appendix gives a simple description of how cells in Zellkalkül are programmed.

The following are two code scripts for the two cellular IDs 0 and 99:

```plaintext
// code if ID 0
for(i=0;i<=1024;i++){
    t=0;
    for(j=1;j<=i;j++){
        if((i%j)==0)
            t++;
    }
    if(t==2){
        writeln(i+" is a prime number.");
        createCell((i%32)-16,Math.floor(i/32)-16,5,99,1);
    }
    else{
        writeln(i+" is not a prime number.");
    }
}
setActive(false);

// code of ID 99
split(4,81,1);
setActive(false);
```

The two code scripts are entered into coding editors, which open after the respective ID colour codes at the bottom of the Zellkalkül user interface are clicked.
In this case, the two colour codes corresponding to IDs 0 and 99 must be clicked successively to open two code editors, into which the two code scripts are entered. Then, execution is started by clicking the VCR-like “play” arrow at the top right corner of the user interface. The initial zygote is always at location 0, 0, 0 and of ID 0 and its code script (the upper one shown above) will be executed. This code tests the numbers between 0 and 1024 for prime numbers and creates cells of ID 99 accordingly in one layer at z=5 above the zygote in the Zellkalkül universe. The last line sets the zygote to passive mode. Therefore, its code will not be executed in future execution cycles any more unless code in another cell re-activates it. In the following execution cycle each of the new cells of ID 99 will perform their code script (the second one shown above), which instructs each to split once into IDs 4 and 81 before setting itself to passive mode, excluding itself from future execution cycles. The next execution cycle will not find any cells in the universe which contain code and are marked as active since the zygote has set itself to passive in the first execution cycle and all remaining cells are of IDs 4 and 81 for which no code script was provided. Figure C.1 shows the user interface before execution start (left), after the first execution cycle (middle) and after the second execution cycle (right).

![Figure C.1: Execution of Zellkalkül example code.](image.png)
This appendix describes the modelling of two examples of pattern formation described in developmental biology. The exercises were carried out using Zellkalkül as part of an attempt to understand the principles by which cellular structures can grow and multiply to express form according to initial intentions based on programming scripts executed by individual cells. They were also discussed in Fischer et al. [2002].

Development from initial zygotes to multi-cellular organisms by means of cellular proliferation and differentiation can proceed by two basic cellular coordination strategies. Sidney Brenner, a pioneer in the field of developmental biology, has (according to Gehring [1998], p. 58) designated them the “American” and the “European” way. Cells developing the American way often “may not even know” their ancestors and draw their sense of identity from interaction with
their neighbors. The European way is for the cells “to do their own thing, not
to talk to their neighbors very much” and to draw their sense of identity from
their developmental history. Both strategies oftentimes appear together in some
combination. They are known to exist in pure form in the development of the
eye discs of the fruit fly *Drosophila melanogaster* (using the American way) and
in the larval development of the nematode worm *Caenorhabditis elegans* (using
the European way). I carried out the following two exercises modelling the two
examples of cellular development within the *Zellkalkül* environment.

The compound eyes of the fruit fly *Drosophila melanogaster* are each composed
of 800 hexagonally packed light-sensing facets called ommatidia, which are each
composed of 20 cells. The individual differentiation of these cells is a well-
understood example of development by local identity in the “American way”
using short-range intercellular communication. After metamorphosis, each of
*Drosophila*’s two eye discs consists of a yet undifferentiated “equivalence group”
of about 20,000 fully identical cells. These identical cells differentiate into a
clear repetitive structure of seven different types of cells, which perform different
functions. Two of these cell-types are so-called cone cells, which secrete an optical
lens for each ommatidium, and five are photoreceptor types, the actual light
sensors. The interesting developmental question here is: How can initially identical
cells develop into a pattern of specialised cells?

The process of differentiation in *Drosophila melanogaster*’s ommatidia (as de-
scribed bu Hafen [1991] and Gerhard and Kirschner [1997], pp. 268 ff.) is initiated
by a hormon wave that sweeps across each eye disc. Figure D.1 shows this
wave (dark vertical band) at the halfway point of its way from the back to the
front side of a disc. While moving, this signaling wave stimulates every cell it
makes contact with to differentiate into a photoreceptor cell named R8. Once a
cell is R8, it prevents the cells surrounding it from becoming R8. As a result, R8 cells differentiate in not-too-close proximity to each other, each surrounded by a hexagonal neighborhood of cells it has prevented from becoming R8s. Each of these neighborhoods develops into one ommatidium with twenty cells. This requires the six cells directly adjacent to each R8 cell to differentiate into photoreceptor cells R1 through R6, in a subsequent step. Then, of the remaining 12 cells that form the outer ring, a group of five again identical cells differentiates into four transparent cone cells (acting as an optical lens) plus an eighth photoreceptor cell R7. R8 sends a shortrange signal to its immediate neighbors. This message can instruct any cell of the equivalence group of five to differentiate into R7. But since only one of the five cells has immediate physical contact with R8, only this one will differentiate into the last photoreceptor cell R7 while the other four do not receive R8’s message and differentiate into cone cells. Two Drosophila mutants exist, both of which fail to develop R7 cells in their ommatidia. One of them cannot produce the message that is sent from R8 to R7 and the other one’s R7s are unable to evaluate R8’s message. In both cases, the ommatidia develop with five cone cells and no R7 cell. One of the two mutants is therefore called “sevenless”.

Figure D.1: Eye-disk development in Drosophila melanogaster. Left: Microscopic view reproduced from Gerhard and Kirschner [1997], color plate 36. Right: Schematic view.

Figure D.2 shows the simulated proliferation of an initial cell (1) into an equivalence group of cells (2) in Zellkäul. According to the equivalence group’s
programming, this tissue develops into the wild type (3) or the “sevenless” mutant (4) with an additional cone cell (bright color) in each ommatidium. The images on the left show microscopic images (reproduced from Hafen [1996]) of the fully developed eye (A), wild type (B) and “sevenless” mutant ommatidia (C).

Figure D.2: Eye-disk development in Drosophila melanogaster in Zellkalkül.

The nematode worm Caenorhabditis elegans represents an example of the “European way” of cellular development. There are two different sexes in this species: hermaphrodites, capable of self-fertilisation and less frequent males. C. elegans has a length of about 1 millimeter and always consists of a total number of 959 somatic cells in the adult hermaphrodite and 1,031 in the adult male plus a variable number of germ cells. Caenorhabditis elegans’ cell lineage (the route by which all cells of the adult are derived from the zygote by cell division, see figure 4) is essentially invariant from organism to organism (see Gehring [1998], pp. 56 ff.).

Every worm develops its lineage of 959 (or 1,031) cells with a very high degree of precision. The identity of a cell, determining its fate (division, tissue expression, programmed death etc.) is a result of the cell’s position in the nematode’s cell lineage tree – the invariant temporal and spatial record of cell divisions.

This model of a cellular developmental process is based on C. elegans’ known
lineage tree\(^1\). Programming an “artificial” zygote with the known lineage history information of the natural worm allowed “growing” a corresponding organism of 959 (or 1,031) cells. This simulation generates a cell cluster based on *C. elegans*’ lineage tree data and a pressure/adhesion optimisation for cell proliferation. Given a cell lineage data set including spatial distribution and cell migration data, I developed a three-dimensional model reproducing the combination of cells present in *C. elegans*’ larva in Zellkalkül.

\(^1\)I have extracted the required data structure from an online data base at http://www.wormbase.com.
This appendix contains analyses of the three types of space-filling polyhedra supported by the Soap Bubble Tool discussed in this thesis: rhombic dodecahedra in section E.1, truncated octahedra (which are the straight-edged basis of so-called Kelvin Cells) in section E.2 and straight-edged variants of the two different polyhedra forming Weaire-Phelan foam. The latter two are 1. an irregular pentagonal dodecahedron, which is discussed in section E.3, and 2. an irregular tetrakaidecahedron with two hexagonal and twelve pentagonal faces, which is discussed in section E.4. Data describing Weaire-Phelan foam is usually obtained by using the Surface Evolver software (see Brakke [1992]). This results in monodisperse foam with curved edges and slightly inaccurate vertex coordinate data. The vertex data shown here has been idealised into a simpler, straight-edged geometric form to allow precise geometry rationalisation without strut curvature. The below data
describes the topology of all polyhedra at unit scale starting with a list of Cartesian coordinates of vertex points followed by a list of edge connections referring to the vertex points. Then, a list of faces refers to edges of the preceding list in counterclockwise direction (from the outside-in view). Negative number references to edges in the face data lists indicate reverse edge direction. In all three packing types each cell can be uniquely addressed in three-dimensional space by one set of three integer coordinates \((x, y, z)\). Vice versa, every set of three integer co-ordinates maps onto exactly one cell location in all three packing types. This common property allows sphere-packing based form finding in the *Soap Bubble Tool* software by using all three packing types via the same three-dimensional packing interface. It also allows the translation of each of the three packing types into each of the two remaining ones. For this purpose, each packing type requires its own conversion function, which converts logical integer \(x\), \(y\) and \(z\) addresses into packing-type-specific \(x\), \(y\) and \(z\) coordinates of centre point locations. Each of the following analyses concludes with its respective conversion function in pseudo-code format. Section E.3 also contains a function in pseudo-code format for determining the choice between both Weaire-Phelan polyhedra within a packed structure based on logical integer \(x\), \(y\) and \(z\) addresses.
E.1 RHOMBIC DODECAHERDON

**Figure E.1:** Vertices in rhombic dodecahedron.

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.707106781</td>
</tr>
<tr>
<td>2</td>
<td>-0.5</td>
<td>0</td>
<td>0.353553391</td>
</tr>
<tr>
<td>3</td>
<td>-0.5</td>
<td>-0.5</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>-0.5</td>
<td>0.353553391</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>-0.5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.5</td>
<td>0</td>
<td>0.353553391</td>
</tr>
<tr>
<td>7</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.5</td>
<td>0.353553391</td>
</tr>
<tr>
<td>9</td>
<td>-0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>-0.5</td>
<td>0</td>
<td>-0.353553391</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>-0.5</td>
<td>-0.353553391</td>
</tr>
<tr>
<td>12</td>
<td>0.5</td>
<td>0</td>
<td>-0.353553391</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.5</td>
<td>-0.353553391</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0</td>
<td>-0.707106781</td>
</tr>
</tbody>
</table>

**Table E.1:** Vertex coordinates in rhombic dodecahedron.
**Figure E.2:** Edges in rhombic dodecahedron.

**Table E.2:** Edge connections in rhombic dodecahedron.
Figure E.3: Faces in rhombic dodecahedron.

Table E.3: Faces in rhombic dodecahedron.

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>-7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>-10</td>
<td>11</td>
<td>12</td>
<td>-1</td>
</tr>
<tr>
<td>5</td>
<td>-12</td>
<td>13</td>
<td>14</td>
<td>-2</td>
</tr>
<tr>
<td>6</td>
<td>-5</td>
<td>15</td>
<td>16</td>
<td>-5</td>
</tr>
<tr>
<td>7</td>
<td>-6</td>
<td>17</td>
<td>18</td>
<td>-8</td>
</tr>
<tr>
<td>8</td>
<td>-9</td>
<td>19</td>
<td>20</td>
<td>-11</td>
</tr>
<tr>
<td>9</td>
<td>-14</td>
<td>21</td>
<td>22</td>
<td>-15</td>
</tr>
<tr>
<td>10</td>
<td>-16</td>
<td>-22</td>
<td>23</td>
<td>-17</td>
</tr>
<tr>
<td>11</td>
<td>-18</td>
<td>-23</td>
<td>24</td>
<td>-19</td>
</tr>
<tr>
<td>12</td>
<td>-20</td>
<td>-24</td>
<td>-21</td>
<td>-13</td>
</tr>
</tbody>
</table>

Address to location conversion for rhombic dodecahedra packing (pseudocode)

```plaintext
placer(xaddress, yaddress, zaddress)
    If zaddress * 0.5 = integer(zaddress / 2) Then
        xlocation = xaddress
        ylocation = yaddress
    Else
        xlocation = xaddress + 0.5
        ylocation = yaddress + 0.5
        zlocation = zaddress * Sqr(2) * 0.5
    return xlocation, ylocation, zlocation
```

284
E.2 TRUNCATED OCTAHEDRON (KELVIN CELL)

Figure E.4: Vertices in truncated octahedron.

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.204124145</td>
<td>0</td>
<td>0.612372436</td>
</tr>
<tr>
<td>2</td>
<td>0.204124145</td>
<td>0</td>
<td>0.612372436</td>
</tr>
<tr>
<td>3</td>
<td>-0.40824829</td>
<td>-0.288675135</td>
<td>0.40824829</td>
</tr>
<tr>
<td>4</td>
<td>0.40824829</td>
<td>-0.288675135</td>
<td>0.40824829</td>
</tr>
<tr>
<td>5</td>
<td>0.40824829</td>
<td>0.288675135</td>
<td>0.40824829</td>
</tr>
<tr>
<td>6</td>
<td>-0.40824829</td>
<td>0.288675135</td>
<td>0.40824829</td>
</tr>
<tr>
<td>7</td>
<td>-0.612372436</td>
<td>0</td>
<td>0.204124145</td>
</tr>
<tr>
<td>8</td>
<td>-0.204124145</td>
<td>-0.577350269</td>
<td>0.204124145</td>
</tr>
<tr>
<td>9</td>
<td>0.204124145</td>
<td>-0.577350269</td>
<td>0.204124145</td>
</tr>
<tr>
<td>10</td>
<td>0.612372436</td>
<td>0</td>
<td>0.204124145</td>
</tr>
<tr>
<td>11</td>
<td>0.204124145</td>
<td>0.577350269</td>
<td>0.204124145</td>
</tr>
<tr>
<td>12</td>
<td>-0.204124145</td>
<td>0.577350269</td>
<td>0.204124145</td>
</tr>
<tr>
<td>13</td>
<td>-0.612372436</td>
<td>0</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>14</td>
<td>-0.204124145</td>
<td>-0.577350269</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>15</td>
<td>0.204124145</td>
<td>-0.577350269</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>16</td>
<td>0.612372436</td>
<td>0</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>17</td>
<td>0.204124145</td>
<td>0.577350269</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>18</td>
<td>-0.204124145</td>
<td>0.577350269</td>
<td>-0.204124145</td>
</tr>
<tr>
<td>19</td>
<td>-0.40824829</td>
<td>0.288675135</td>
<td>-0.40824829</td>
</tr>
<tr>
<td>20</td>
<td>0.40824829</td>
<td>0.288675135</td>
<td>-0.40824829</td>
</tr>
<tr>
<td>21</td>
<td>0.40824829</td>
<td>-0.288675135</td>
<td>-0.40824829</td>
</tr>
<tr>
<td>22</td>
<td>-0.40824829</td>
<td>-0.288675135</td>
<td>-0.40824829</td>
</tr>
<tr>
<td>23</td>
<td>-0.204124145</td>
<td>0</td>
<td>-0.612372436</td>
</tr>
<tr>
<td>24</td>
<td>0.204124145</td>
<td>0</td>
<td>-0.612372436</td>
</tr>
</tbody>
</table>

Table E.4: Vertex coordinates in truncated octahedron.
Figure E.5: Edges in truncated octahedron.

Table E.5: Edge connections in truncated octahedron.
Figure E.6: Faces in truncated octahedron.

Table E.6: Faces in truncated octahedron.

Address to location conversion for truncated octahedra packing (pseudocode)

```pseudocode
placer(xaddress, yaddress, zaddress)
xlocation = inarr(0) * 0.81649658
ylocation = inarr(1) * 1.154700538
zlocation = inarr(2) * 0.81649658
If xaddress * 0.5 = integer(xaddress / 2) And Not yaddress * 0.5 = integer(yaddress / 2) Then
    ylocation = ylocation + 0.577350269
Else
    ylocation = ylocation + 0.577350269
End If
If Not xaddress * 0.5 = integer(xaddress / 2) And yaddress * 0.5 = integer(yaddress / 2) Then
    ylocation = ylocation + 0.577350269
Else
    ylocation = ylocation + 0.577350269
End If
return xlocation, ylocation, zlocation
```
### E.3 WP-LIKE PENTAGONAL DODECAHEDRON

**Figure E.7:** Vertices in Weaire-Phelan-like pentagonal dodecahedron.

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.333333333</td>
<td>0.666666667</td>
</tr>
<tr>
<td>2</td>
<td>-0.444444444</td>
<td>0.444444444</td>
<td>0.444444444</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-0.333333333</td>
<td>0.666666667</td>
</tr>
<tr>
<td>4</td>
<td>0.444444444</td>
<td>0.444444444</td>
<td>0.444444444</td>
</tr>
<tr>
<td>5</td>
<td>-0.666666667</td>
<td>0</td>
<td>0.333333333</td>
</tr>
<tr>
<td>6</td>
<td>-0.444444444</td>
<td>-0.444444444</td>
<td>0.444444444</td>
</tr>
<tr>
<td>7</td>
<td>0.444444444</td>
<td>-0.444444444</td>
<td>0.444444444</td>
</tr>
<tr>
<td>8</td>
<td>0.666666667</td>
<td>0</td>
<td>0.333333333</td>
</tr>
<tr>
<td>9</td>
<td>0.333333333</td>
<td>0.666666667</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>-0.333333333</td>
<td>0.666666667</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>-0.444444444</td>
<td>0.444444444</td>
<td>-0.444444444</td>
</tr>
<tr>
<td>12</td>
<td>0.333333333</td>
<td>-0.666666667</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0.666666667</td>
<td>0</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>14</td>
<td>0.444444444</td>
<td>0.444444444</td>
<td>-0.444444444</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.333333333</td>
<td>-0.666666667</td>
</tr>
<tr>
<td>16</td>
<td>-0.666666667</td>
<td>0</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>17</td>
<td>-0.333333333</td>
<td>-0.666666667</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0.444444444</td>
<td>-0.444444444</td>
<td>-0.444444444</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>-0.333333333</td>
<td>-0.666666667</td>
</tr>
<tr>
<td>20</td>
<td>-0.444444444</td>
<td>-0.444444444</td>
<td>-0.444444444</td>
</tr>
</tbody>
</table>

**Table E.7:** Vertex coordinates in Weaire-Phelan-like pentagonal dodecahedron.
Figure E.8: Edges in Weaire-Phelan-like pentagonal dodecahedron.

Table E.8: Edge connections in Weaire-Phelan-like pentagonal dodecahedron.
Figure E.9: Faces in Weaire-Phelan-like pentagonal dodecahedron.

Table E.9: Faces in Weaire-Phelan-like pentagonal dodecahedron.
Address to location conversion for Weaire-Phelan-like packing (pseudocode)

```plaintext
placer(xaddress, yaddress, zaddress)
    If zaddress / 2 = integer(zaddress / 2) Then
        If xaddress / 2 = integer(xaddress / 2) Then
            If yaddress / 2 = integer(yaddress / 2) Then
                xlocation = xaddress - 0.125
                ylocation = yaddress + 0.125
                zlocation = zaddress - 0.5
            Else
                xlocation = xaddress - 0.125
                ylocation = yaddress + 0.625
                zlocation = zaddress - 0.5
            Else
                xlocation = xaddress - 0.125
                ylocation = yaddress + 0.625
                zlocation = zaddress
            Else
                If yaddress / 2 = integer(yaddress / 2) Then
                    xlocation = xaddress + 0.375
                    ylocation = yaddress + 0.125
                    zlocation = zaddress - 0.5
                Else
                    xlocation = xaddress - 0.125
                    ylocation = yaddress + 0.125
                    zlocation = zaddress
                Else
                    If yaddress / 2 = integer(yaddress / 2) Then
                        xlocation = xaddress + 0.375
                        ylocation = yaddress + 0.125
                        zlocation = zaddress - 0.5
                    Else
                        xlocation = xaddress - 0.125
                        ylocation = yaddress + 0.125
                        zlocation = zaddress - 0.5
                    Else
                        xlocation = xaddress - 0.125
                        ylocation = yaddress + 0.125
                        zlocation = zaddress
                    Else
                        If xaddress / 2 = integer(xaddress / 2) Then
                            If yaddress / 2 = integer(yaddress / 2) Then
                                xlocation = xaddress + 0.375
                                ylocation = yaddress + 0.125
                                zlocation = zaddress - 0.5
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress - 0.5
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddress
                            Else
                                xlocation = xaddress - 0.125
                                ylocation = yaddress + 0.125
                                zlocation = zaddres
APPENDIX E - POLYHEDRA GEOMETRY AND PACKING ANALYSES

Polyhedron choice for Weaire-Phelan-like packing

```plaintext
polyhedronchooser(xaddress, yaddress, zaddress)
  If zaddress / 2 = Int(zaddress / 2) Then
    If xaddress / 2 = Int(xaddress / 2) Then
      If yaddress / 2 = Int(yaddress / 2) Then // z even, x even, y even
        chose pentagonal dodecahedron in basic rotation,
        edge 1 parallel to y-axis
      Else // x even, y odd, z even
        chose tetrakaidecahedron, rotated around x-axis
        by 90 degrees, face 7 facing into z-direction
    Else // y even, x odd, z even
      chose tetrakaidecahedron in basic rotation,
      face 7 facing into -y direction
    Else // y odd, x even, z even
      chose tetrakaidecahedron, rotated 90 degrees
      around x-axis, face 7 facing into z-direction
  Else // x odd, y even, z even
    chose tetrakaidecahedron, rotated 180 degrees
    around x-axis, face 7 facing into y-direction
  Else // x odd, y odd, z even
    chose tetrakaidecahedron, rotated 90 degrees
    around z-axis, face 7 facing into -x-direction
  Else // z odd, x even, y even
    chose tetrakaidecahedron, rotated 90 degrees
    around z-axis, face 7 facing into -z-direction
  Else // z odd, x odd, y even
    chose tetrakaidecahedron, rotated 90 degrees
    around x-axis, face 7 facing into -z-direction
  Else // z odd, x odd, y odd
    chose pentagonal dodecahedron, rotated around
    z-axis by 90 degrees, edge 1 parallel to x-axis
```

292
E.4 WEaire-PheLAN-Like Tetrakaidecahedron

Figure E.10: Vertices in Weaire-Phelan-like tetrakaidecahedron.

Table E.10: Vertex coordinates in Weaire-Phelan-like tetrakaidecahedron.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0</td>
<td>-0.166666667</td>
<td>0.666666667</td>
</tr>
<tr>
<td>2 -0.555555556</td>
<td>-0.055555556</td>
<td>0.444444444</td>
</tr>
<tr>
<td>3 0</td>
<td>-0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4 0.444444444</td>
<td>0.055555556</td>
<td>0.444444444</td>
</tr>
<tr>
<td>5 0.444444444</td>
<td>0.055555556</td>
<td>0.555555556</td>
</tr>
<tr>
<td>6 -0.444444444</td>
<td>0.055555556</td>
<td>0.555555556</td>
</tr>
<tr>
<td>7 -0.333333333</td>
<td>-0.5</td>
<td>0.333333333</td>
</tr>
<tr>
<td>8 0.333333333</td>
<td>-0.5</td>
<td>0.333333333</td>
</tr>
<tr>
<td>9 0.333333333</td>
<td>0.5</td>
<td>0.333333333</td>
</tr>
<tr>
<td>10 -0.333333333</td>
<td>0.5</td>
<td>0.333333333</td>
</tr>
<tr>
<td>11 -0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>12 0.5</td>
<td>-0.5</td>
<td>0</td>
</tr>
<tr>
<td>13 0.666666667</td>
<td>0.166666667</td>
<td>0</td>
</tr>
<tr>
<td>14 -0.666666667</td>
<td>0.166666667</td>
<td>0</td>
</tr>
<tr>
<td>15 0.333333333</td>
<td>0.5</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>16 0.333333333</td>
<td>0.5</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>17 -0.555555556</td>
<td>-0.055555556</td>
<td>-0.444444444</td>
</tr>
<tr>
<td>18 -0.333333333</td>
<td>-0.5</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>19 0.333333333</td>
<td>-0.5</td>
<td>-0.333333333</td>
</tr>
<tr>
<td>20 0.555555556</td>
<td>-0.055555556</td>
<td>-0.444444444</td>
</tr>
<tr>
<td>21 0.444444444</td>
<td>0.055555556</td>
<td>-0.555555556</td>
</tr>
<tr>
<td>22 0.5</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>23 -0.444444444</td>
<td>0.055555556</td>
<td>-0.555555556</td>
</tr>
<tr>
<td>24 0.5</td>
<td>-0.166666667</td>
<td>-0.666666667</td>
</tr>
</tbody>
</table>
Figure E.11: Edges in Weaire-Phelan-like tetrakaidecahedron.

Table E.11: Edge connections in Weaire-Phelan-like tetrakaidecahedron.
Figure E.12: Faces in Weaire-Phelan-like tetrakaidecahedron.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>-4</td>
<td>5</td>
<td>-6</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>-8</td>
<td>9</td>
<td>-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>10</td>
<td>17</td>
<td>-11</td>
<td>-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>-19</td>
<td>-18</td>
<td>-11</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-9</td>
<td>15</td>
<td>-20</td>
<td>-16</td>
<td>-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>21</td>
<td>-26</td>
<td>-13</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>13</td>
<td>31</td>
<td>-32</td>
<td>-14</td>
<td>-7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>28</td>
<td>-24</td>
<td>-15</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>16</td>
<td>23</td>
<td>-25</td>
<td>-22</td>
<td>-18</td>
<td>-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>30</td>
<td>-26</td>
<td>-22</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>24</td>
<td>33</td>
<td>-27</td>
<td>-23</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>30</td>
<td>34</td>
<td>-36</td>
<td>-31</td>
<td>-29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>36</td>
<td>-35</td>
<td>-33</td>
<td>-28</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>27</td>
<td>35</td>
<td>-34</td>
<td>-26</td>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table E.12: Faces in Weaire-Phelan-like tetrakaidecahedron.

Address to location conversion and polyhedron choice: See end of section E.3
Appendix F

Tsunami Memorial Design Competition Submission

This appendix contains panels submitted to the Tsunami Memorial Design Competition in Thailand in 2005. Figures F.1 and F.2 are the submission by Chris Bosse (PTW), Tristram Carfrae and Stuart Bull (Arup) and myself to the competition stage I. Figures F.3 and F.4 are the stage I submission by the design team around Ana Somoza Jimenez, who eventually won the competition. In their feedback to the shortlisted designers following stage I the competition jury commented on this entry: “These organic, tower-like structures, slightly reminiscent of pagodas, were thought to be unbounded by any particular reference to time or space and therefore provide a symbolic landmark while integrating with the natural features of the site. Parking spaces, pathway access to the beach and the form of the exhibition spaces need to be made clear. More clarity is needed about the structure of the towers and research needs to be made into the climatic and topographical conditions of the site in relation to the structure.”¹ Figures F.5 through F.9 show the panels submitted by the team around Ana Somoza Jimenez at competition stage II.

¹See http://www.tsunamimemorial.or.th/finalist1.htm
**Figure F.1:** First poster panel entered into *Tsunami Memorial Design Competition* stage I.
Figure F.2: Second poster panel entered into Tsunami Memorial Design Competition stage I.
Figure F.3: First poster panel entered into Tsunami Memorial Design Competition stage I by Ana Somoza Jimenez et al.
Figure F.4: Second poster panel entered into Tsunami Memorial Design Competition stage I by Ana Somoza Jimenez et al.
Figure F.5: First poster panel entered into Tsunami Memorial Design Competition stage II by Ana Somoza Jimenez et al.
Figure F.6: Second poster panel entered into Tsunami Memorial Design Competition stage II by Ana Somoza Jimenez et al.
Figure F.7: Third poster panel entered into Tsunami Memorial Design Competition stage II by Ana Somoza Jimenez et al.
Figure F.8: Forth poster panel entered into Tsunami Memorial Design Competition stage II by Ana Somoza Jimenez et al.
Figure F.9: Fifth poster panel entered into *Tsunami Memorial Design Competition* stage II by Ana Somoza Jimenez et al.
This appendix recounts how the soap bubble tool was used during the *Tsunami Memorial Design Competition* (see section 3.11) in the form of screen shots of key stages. Figure G.1 shows the basic setup with the *Rhino3D®* modelling package in the top right window and an *MS Excel®* spreadsheet in the top left window. The code editor in the bottom window is the *MS Excel®* VBA script editor with some tool code visible. This script is the code of the actual tool interface visible as a smaller window in the lower right-hand side of the screen. It is associated with the spreadsheet and remote-controls the *Rhino3D®* package. The tool interface has several tabbed pages, allowing access to different functions in the sequence of their typical use. The page shown in figure G.1 contains what could be called the “bubble cluster editor” in the lower right part. This interface allows the construction of packed spheres, which may later be converted into a grid structure circumscribing the bubbles. The red circle visible in *Rhino3D®* is the tool’s “bubble cursor”. This cursor can be moved in three dimensions from the shown tool interface and indicates the location at which a bubble may be added, deleted or modified. The
spreadsheet shown at the top left contains the code of the tool and controls the three-dimensional modelling in *Rhino3D®*. The display of the code editor window at the bottom is optional. Typical use does not involve direct manipulation of the spreadsheet or the code editor.

![Basic soap bubble tool setup.](image1)

**Figure G.1**: Basic soap bubble tool setup.

Figure G.2 shows the same view after a few spheres have been added in the x/y plane using the tools sphere packing-editing functions.

![Sphere clustering in two dimensions.](image2)

**Figure G.2**: Sphere clustering in two dimensions.
Figures G.3 and G.4 show the same view with more spheres added in the third dimension, forming a cluster of packed spheres. The only packing type supported by the tool at the time of our work on the Tsunami Memorial Design Competition was that leading to the expression of grid structures corresponding to packed truncated octahedra (which are the straight-edged basis of so-called Kelvin Cells, see section E.2).

Figure G.3: Sphere clustering in three dimensions.

Figure G.4 shows how on the next tabbed page of the tool “virtual pressures” can be assigned to individual cells for later parametric distortion. The tool’s entry fields for up to six different “virtual pressure” parameters are coloured and spheres are coloured according to the parameters assigned to them.

Figure G.5 shows how after finishing the clustering of spheres and assigning “virtual pressure” parameters, the constructed cluster is first translated into a basic, non-distorted grid structure circumscribing all spheres on the next tabbed page of the tool. This serves as a visual clue during the establishing of the grid topology before parametric alteration.

Figure G.6 shows how after a short while the parametrically altered space
grid is generated. The structure based on packed truncated octahedra remains topologically identical while cell sizes are scaled according to the previously assigned “virtual pressure” parameters. The algorithm for this transformation is discussed in section 3.9 and in particular in figure 3.36.

Figure G.7 shows how functions accessible via the following tabbed page of the tool allows numerical analysis of the resulting space grid structure. The previously assembled colour-coded spheres are hidden and the contained strut lengths are measured, sorted and displayed in the spreadsheet shown in the top left of the shown screen view. Horizontal progressions in the point distribution indicate sets of equal strut lengths. The actual number of different strut types with different lengths necessary for constructing the respective space frame structure depends on the tolerance afforded by the given scale, material and fabrication processes. Greater tolerances decrease the number of strut types.

As shown in figure G.8, the tool has the capability of exporting generated structures as “MAX Script” macros, which can be loaded and executed in 3ds Max® in a range of different expressions. It is possible to select different strut diameters
and material properties and the inclusion of the “soap film” walls (shown in blues colour) can be deactivated. This script-generating interface and the grid expressions supported by it have been under heavy development during the work for the Tsunami Memorial Design Competition as described in greater detail in section 3.10.

Apart from macro output to 3ds Max®, the soap bubble tool was also capable of outputting macros in ANSYS Parametric Design Language (APDL) of the ANSYS® structural modelling package. As shown in figure G.9, the model was loaded with top and side loads in basic stress simulations. Without commitment to concrete scales or materials, this type of simulation was not meant to be realistic or rigorous but intended to demonstrate technical possibilities in the competition entry.
Figure G.6: Sphere packing and derived irregular space grid.

Figure G.7: Removal of spheres and counting of different strut lengths.
APPENDIX G - SOAP BUBBLE TOOL USE DURING DESIGN COMPETITION

Figure G.8: Import into 3ds Max®.

Figure G.9: Import into ANSYS® and structural simulation.
Figure H.1: User interface of the soap bubble tool with added support for packed truncated octahedra and Weaire-Phelan-like structures.
Figure H.2: "Quick-start card" for soap bubble tool (front and back sides).
Figure H.3: "Quick-start card" for tofu cube tool (front and back sides).
# FEEDBACK QUESTIONNAIRE

**TOFU BLOCK**

1. Did you use the Tofu Automata Generator during the workshop?
   - [ ] Yes
   - [ ] No - please give a reason if possible:
     - I do not have a lot of skills in using 3D Studio Max
     - I do not understand the Tofu Automata Generator well enough
     - In my group I was responsible for ___________________________ in stead
     - Other reasons:
     - Please proceed with question 2

2. Did you use the Tofu Automata Generator by yourself or working closely with other group members?
   - [ ] Always by myself
   - [ ] Always working closely with other group members
   - [ ] Sometimes by myself, sometimes working closely with other group members

3. Please describe your reasons for your answer to question 2:

4. Did you start by developing your ideas on paper before using the Tofu Automata Generator on paper or did you use the software from the beginning?
   - [ ] I used paper to develop my initial ideas
   - [ ] Working closely with other group members
   - [ ] I did not use paper sketching I used the Tofu Automata Generator from the beginning
   - [ ] I have used this approach to generate ideas:

5. Does your final Tofu Automata Generator outcome match your initial design ideas? (1=Not at all, 7=Completely)
   - [ ] 1
   - [ ] 2
   - [ ] 3
   - [ ] 4
   - [ ] 5
   - [ ] 6
   - [ ] 7
   - [ ] I don’t know

6. Did you use the Tofu Automata Generator:
   - [ ] 1 = to realise ideas that I had since the beginning of the workshop
   - [ ] 2 = to explore and experiment
   - [ ] 3 = other reason:
     - Please proceed with question 15

7. Did you incorporate surprising results into your design? (1 = No at all, 7 = All of them)
   - [ ] 1
   - [ ] 2
   - [ ] 3
   - [ ] 4
   - [ ] 5
   - [ ] 6
   - [ ] 7
   - [ ] I don’t know

8. Did you use the Tofu Automata Generator to:
   - [ ] 1 = to realise ideas I had since the beginning of the workshop
   - [ ] 2 = to explore and experiment
   - [ ] 3 = other reason:

9. How useful did you find each of the functions of the Tofu Automata Generator? (1 = useless, 7 = very useful)

<table>
<thead>
<tr>
<th>Function</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>I don’t know</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repel</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiply</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orient on Surface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete Intersecting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10. Did you miss additional functions in the Tofu Automata Generator that would have been helpful to you?
    If yes, please describe:

11. Would you have liked to create your own additional functions in the Tofu Automata Generator?
    If yes, please describe:

12. Did you find the four material types (Matter, Void, Context, Neutral) useful? (1 = very hard to understand, 7 = very easy to understand)
    - [ ] 1
    - [ ] 2
    - [ ] 3
    - [ ] 4
    - [ ] 5
    - [ ] 6
    - [ ] 7
    - [ ] I don’t know

13. Did you find the four material types (Matter, Void, Context, Neutral) easy to understand?
    If yes, please describe:

14. Would you have liked to create your own material types in the Tofu Automata Generator?
    If yes, please describe:

---

Figure H.4: Pages 1 and 2 of workshop questionnaire sheet.

APPENDIX H - MATERIAL USED IN THE POSTGRADUATE DESIGN WORKSHOP
15. Did you use the Soap-Bubble Co-Rationaliser during the workshop?

Yes

I do not have a lot of skills in using Excel or Rhino

I do not understand the Soap-Bubble Co-Rationaliser well enough

In my group I was responsible for _____________________________ instead

Other reasons:

Please proceed with question 16

16. Did you use the Soap-Bubble Co-Rationaliser by yourself or working closely with other group members?

Always by myself

Always working closely with other group members

Sometimes by myself, sometimes working closely with other group members

17. Please describe your reasons for your answer to question 2

18. Did you start by developing your ideas on paper before using the Soap-Bubble Co-Rationaliser on paper or did you use this software tool from the beginning?

I used paper to develop my initial ideas

By myself

Working closely with other group members

I did not use paper sketches in the Soap-Bubble Co-Rationaliser from the beginning

I have used this approach to generate ideas:

19. Does your final Soap-Bubble Co-Rationaliser outcome match your initial design ideas? (1 = Not at all, 7 = Completely)

1 2 3 4 5 6 7 8 I don't know

20. Did using the Soap-Bubble Co-Rationaliser inspire you and help you explore new design ideas? (1 = Not at all, 7 = All the time)

1 2 3 4 5 6 7 8 I don't know

21. Did using the Soap-Bubble Co-Rationaliser give you surprising results? (1 = Not at all, 7 = All the time)

1 2 3 4 5 6 7 8 I don't know

Did you incorporate surprising results into your design? (1 = None at all, 7 = All of them)

1 2 3 4 5 6 7 8 I don't know

22. Did you use the Soap-Bubble Co-Rationaliser

1 = to realise ideas I had since the beginning

7 = to explore and experiment

1 2 3 4 5 6 7 I don't know

23. Which type of bubble packing gave you the visually most interesting results?

Packing of rhombic dodecahedra (D)

A packing of Kelvin-like cells (K)

A packing of Weaire-Phelan-like cells (WP)

1 2 3 4 5 6 7 I don't know

24. How useful did you find each of the features of the Soap-Bubble Co-Rationaliser? (1 = Useless, 7 = Very useful)

Cursor control for arranging bubbles

Volumes assignment

Storage containers for variations

1 2 3 4 5 6 7 I don't know

25. Did you miss additional features in the Soap-Bubble Co-Rationaliser that would have been helpful to you?

If yes, please describe:

26. Would you like to add more features in the Soap-Bubble Co-Rationaliser?

If yes, please describe:

GENERAL BLOCK

27. Would you like to make a general comment regarding Generative Design, the software tools used in this workshop or the workshop as a whole?

THANK YOU VERY MUCH!
This appendix lists data collected in a questionnaire that was carried out shortly before the final critique of the postgraduate design workshop discussed in section 3.11. This appendix further lists sample interaction data logged by the Soap Bubble Tool during student’s design process. The full questionnaire comprises 27 questions, of which questions 1-14 and question 27 relate to the Tofu Tool, and questions 15-27 relate to the Soap Bubble Tool. This section lists the results of the data collection.

**Question 1**
Did you use the Tofu Automata Generator during the workshop?
15 students answered this question with “Yes”.
2 students answered this question with “No”. Of these two, only one student gave a reason: “Because this Tofu Automata I think is some like shape grammar we played before. Second, we’re more interested in Bubble Co-Rationaliser.”

**Question 2**
Did you use the Tofu Automata Generator by yourself or working closely with other group members?
1 student answered this question with “Always by myself”.
4 students answered this question with “Always working closely with other group members”.
9 students answered this question with “Sometimes by myself, sometimes working closely with other group members”.
3 students gave no answer to this question.

**Question 3**
Please describe your reasons for your answer to question 2.
Reasons given for the answer “Always by myself”:
“We separate different parts of our work. Everyone has his own system to develop.”
Reasons given for the answer “Always working closely with other group members”:
1. “Because I do not have a lot of skills in using 3DStudio Max. But I’m curious about the software’s effecting work. It looks so different that you even don’t know what it will become.”
2. “We always discuss on how to do and use and it’s very good.”
3. “First time to use. I always need to exchange more experiences with each other.”
4. “Sometimes used at home and sometimes used at school. We discuss together and develop everyone’s ideas.”
Reasons given for the answer “Sometimes by myself, sometimes working closely with other group members”:
1. “At some times, different variance cause the result, so I discuss with my group member. But if the situation that have much factors, I do myself.”
2. “Sometimes when we discuss, we will use the Tofu together. And sometimes I will operate the Tofu by myself because I want to try the ability or setting of Tofu.”
3. “We need discussion about design or the function of software.”
4. “At first I worked with my group members to know how to use the Tofu Automata Generator. After that, for finishing our design on time, we used the Soap-Bubble Co-Rationaliser.”
5. “Need some discussion.”
6. “I have some skills in 3D Studio MAX and someone always asks me about 3DMAX.”
7. “Because sometimes I am busy in other things, and sometimes other group members are busy.”
8. “Because sometimes used at school, sometimes at home.”
9. “In the lab we try to understand the process of generating forms and logic together, and we try to generate different forms by each other.”

**Question 4**
Did you start by developing your ideas on paper before using the Tofu Automata Generator on paper or did you use the software from the beginning?
2 students answered this question with “I used paper to develop my initial ideas”.
5 students answered this question with “I used paper to develop my initial ideas, working closely with other group members”.
6 students answered this question with “I did not use paper sketching. I used the Tofu Automata Generator from the beginning”.

**Question 5**
Does your final Tofu Automata Generator outcome match your initial design ideas?
This question was answered on a scale of 1 = “Not at all” to 7 = “Completely”.
13 students answered this question. The average answer value was 2.92, with a standard derivation of 1.64.
Question 6
Did using the Tofu Automata Generator inspire you and help to explore new design ideas?
This question was answered on a scale of 1 = “Not at all” to 7 = “Always”.
The average answer value was 4.31, with a standard derivation of 2.13.

Question 7
Did using the Tofu Automata Generator give you surprising results?
This question was answered on a scale of 1 = “Not at all” to 7 = “Always”.
The average answer value was 4.615, with a standard derivation of 1.82.
Did you incorporate surprising results into your design?
This question was answered on a scale of 1 = “None at all” to 7 = “All of them”.
The average answer value was 3.384, with a standard derivation of 1.15.

Question 8
Did you use the Tofu Automata Generator to ...?
This question was answered on a scale of 1 = “to realize ideas I had since the beginning of the workshop” to 7 = “to explore and experiment”.
Two clusters of answers can be observed from the answers to this question: one around the value “3” (3 students gave “3”, 1 student gave “4”), and one around the value “6” (5 students gave “6”, 3 students gave “7”).

Question 9:
How useful did you find each of the functions of the Tofu Automata Generator?
This question was answered on a scale of 1 = “Useless” to 7 = “Very useful”, separated by function. The following table illustrates the aggregate values for each question, further distinguished by colours indicating group membership.

Question 10
Did you miss additional functions in the Tofu Automata Generator that would have been helpful to you? If yes, please describe.
No new functions were suggested.

Question 11
Would you have liked to create your own additional functions in the Tofu Automata Generator? If yes, please describe.
5 students answered this question:
1. “I liked to create some functions, such like twist or bend etc.”
2. “Transformation of the box. We can use it to generate more shapes we want.”
3. “Maybe not only the box system, can use the curve line to Multiply or other functions.”
4. “The Tofu software has the potential ability to become an analysis system to plan the layout.”
5. “Yes. Maybe it should have a “mix” function, which can combine different functions.”
Question 12
Did you find the four material types (Matter, Void, Context, Neutral) useful? This question was answered on a scale of 1 = “Useless” to 7 = “Very useful”. Two clusters of answers can be observed from the answers to this question: one around the value “3” (3 students gave “3”, 2 student gave “2”), and one around the value “6” (5 students gave “6”, 2 students gave “5” and 1 student gave “7”).

Question 13
Did you find the four material types (Matter, Void, Context, Neutral) easy to understand?
This question was answered on a scale of 1 = “Very hard to understand” to 7 = “Very easy to understand”.
Two clusters of answers can be observed from the answers to this question: a small cluster around the value “2” (3 students), and one around the value “6” (3 students gave “6”, 3 students gave “5” and 3 student gave “7”).

Question 14
Would you have liked to create your own material types in the Tofu Automata Generator? If yes, please describe.
4 students gave suggestions:
1. “Yes, maybe more realistic material (weight, time?)…”
2. “I think the material types of Tofu are very useful and enough! If we create too many functions, we may get confused.”
3. “More material can help me to define complex system that I would like to make.”
4. “No, I feel the Tofu has enough varieties that I need more time to understand and try for a longer time.”

Question 15
Did you use the Soap Bubble Co-Rationaliser during the workshop?
15 students answered this question with “Yes”.
2 students answered this question with “No”. The reasons given were: “In my group I was responsible for the bookshelf system instead (I used the Tofu Automata Generator to develop the system).” The second reason was: “We have no time, but we think it would be useful to develop our bookshelf shapes.”

Question 16
Did you use the Soap-Bubble Co-Rationaliser by yourself or working closely with other group members?
0 students answered this question with “Always by myself”.
7 students answered this question with “Always working closely with other group members”.
7 students answered this question with “Sometimes by myself, sometimes working closely with other group members”.
2 students gave no answer to this question.

Question 17
Please describe your reasons for your answer to question 2.
Reasons given for the answer “Always working closely with other group members”:
1. “I can’t see exactly where the bubble is positioned, and I need others to remind me.”
2. “We used it in different places. At school we work together, if at home we develop by ourselves.”
3. “I want to discuss how the soap-bubbles can be used in the design.”
4. “The software is easy to create new organic sphere structures - like nest of bees. And it’s interesting to operate.”
5. “I’m learning 3DSMAX now.”
6. “Different results by our group can prepare different factors. What make the results.”
Reasons given for the answer “Sometimes by myself, sometimes working closely with other group members”:
1. “The Soap-Bubble Co-Rationaliser is an interesting software; we often discuss about it!”
2. “I want to find something new to me and share with my team members.”
3. “We used the Soap-Bubble Co-Rationaliser to develop some elements, like covering, furniture....sometimes we used it separately (or developed covering, the other developed to discuss furniture) and sometimes we worked together.”
4. “We used ‘soap’ to generate different elements. I was responsible for some element and then merged with other elements made by other group members.”
5. “It’s a dynamic process, that it surprises you in the design process. But it is sometimes too various that you would be a bit confused to use it.”

**Question 18**
Did you start by developing your ideas on paper before using the Soap-Bubble Co-Rationaliser on paper or did you use this software tool from the beginning?
1 student answered this question with “I used paper to develop my initial ideas”.
6 students answered this question with “I used paper to develop my initial ideas, working closely with other group members”.
7 students answered this question with “I did not use paper sketching. I used the Soap-Bubble Co-Rationaliser from the beginning”.
1 student answered this question with “I have used this approach to generate ideas: With the forms-developing of intention to set the position and then to replace of the new object.”
2 students gave no answer to this question.

**Question 19**
Does your final Soap Bubble Co-Rationaliser outcome match your initial design ideas?
This question was answered on a scale of 1 = “Not at all” to 7 = “Completely”.
15 students answered this question. The average rating was 5.13, with a standard deviation of 1.93.

**Question 20**
Did using the Soap Bubble Co-Rationaliser inspire you and help to explore new
design ideas? This question was answered on a scale of 1 = “Not at all” to 7 = “Always”. 15 students answered this question. The average answer value was 5.73, with a standard deviation of 0.57.

**Question 21**
Did using the Soap-Bubble Co-Rationaliser give you surprising results?
This question was answered on a scale of 1 = “Not at all” to 7 = “Always”. 15 students answered this question. The average answer value was 5.53, with a standard derivation of 0.96. Did you incorporate surprising results into your design?
This question was answered on a scale of 1 = “None at all” to 7 = “All of them”. 15 students answered this question. The average answer value was 4.67, with a standard derivation of 1.14.

**Question 22**
Did you use the Soap-Bubble Co-Rationaliser to ...?
This question was answered on a scale of 1 = “to realize ideas I had since the beginning of the workshop” to 7 = “to explore and experiment”.
14 students answered this question. The average answer value was 5.5, with a standard derivation of 1.5.

**Question 23**
Which type of bubble packing gave you the visually most interesting results?
This question was answered by choosing between three options: Packing of rhombic dodecahedra (D), Packing of Kelvin-like cells (K), or Packing of Weaire-Phelan-like cells (WP).
0 students answered this question with “D”
1 student answered this question with “K”
13 students answered this question with “WP”
3 students gave no answer to this question.

**Question 24**
How useful did you find each of the features of the Soap-Bubble Co-Rationaliser?
This question was answered on a scale of 1 = “Useless” to 7 = “Very useful”, separated by function. The following table illustrates the aggregate values for each question, further distinguished by colours indicating group membership.

**Question 25**
Did you miss additional functions in the Soap Bubble Co-Rationaliser that would have been helpful to you? If yes, please describe.
6 students answered this question:
1. “The cursor control is hard to see when I have made some bubbles.”
2. “Yes, it can easily create a complex form.”
3. “I very much like the software.”
4. “I like the result of its shape. The structure looks organic and interesting.”
5. “Box filler. It can create some interesting shape.”
6. “Box filler, because it can make the bubbles consist in a clear shape.”

**Question 26**
Would you have liked to create your own additional functions in the Soap Bubble Co-Rationaliser? If yes, please describe.
5 students answered this question:
1. “Yes, if it can be used in structure maybe useful.”
2. “That’s a useful software, and inspires my ideas!”
3. “I liked to play with forms like this (sphere-like) weaved by bamboo sticks.”
4. “Maybe you can add some functions to create more organic shapes like (spirals, loops, curves).”
5. “Maybe use different definition of？ Maybe it’s a different mathematic rules to help construction or connect to the constraints of reality.”

**Question 27**
Would you like to make a general comment regarding Generative Design, the software tools used in this workshop or the workshop as a whole?
11 students gave suggestions:
1. “Actually, I like architecture as simple as possible. But I think the Generative Design is helpful for constructing, because the whole development is based on a simple unit (like Tofu or Bubble) and it is much easy to control.”
2. “In the workshop, we try to define the scale meaning and how to match two systems and the site. And then we used that to create the rules is interested.”
3. “I don’t think Generative Design can design because there are many space elements can’t in to computer, like nature. So the generative Design is a very good ‘tool’.”
4. “Yes, the software tools are really interesting and inspiring! Of course, it’s good to think about generative Design. That’s a nice experiment to discuss about Design during the workshop. It’s pretty good!”
5. “I like the Soap-Bubble tool because of it’s a nice form and magic spaces producer.”
6. “Of course, the digital tool is interesting. The workshop is pretty good.”
7. “Maybe you can add some functions to create more organic shapes like (spirals, loops, curves).”
8. “Yes, I’d like to use for original form development. To filler until the used ones, and helps me to program new ideas in real uses.”
9. “I think the designer using the software tools for Generative Design must have very powerful skill for the software. Otherwise they must test the software again and again to experience is the result what the designer wants.”
10. “In general, we will use Tofu software to develop some unexpected forms in Generative Design. But I think the Tofu software can be used to be a useful analytic system. It can test the ability of layout based on different contexts in the site planning.”
11. “It’s very useful to student and the agenda is in normal speed. If could, maybe we can have several workshops like this one and help us improve our vision and make a different trainings.”
Sample page of tool logging data.

<table>
<thead>
<tr>
<th>Date and Time</th>
<th>Host ID</th>
<th>Tool</th>
<th>Group</th>
<th>Command</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/10/2006</td>
<td>ASUS_08</td>
<td>soap</td>
<td>addbubble</td>
<td>3 lines 17</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4/10/2006</td>
<td>ASUS_08</td>
<td>soap</td>
<td>addbubble</td>
<td>3 lines 17</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4/10/2006</td>
<td>ASUS_08</td>
<td>soap</td>
<td>delbubbles</td>
<td>delallbubbles</td>
<td>1 1 1</td>
</tr>
<tr>
<td>4/10/2006</td>
<td>ASUS_08</td>
<td>soap</td>
<td>fillbubbles</td>
<td>fillbubbles</td>
<td>0 0 0</td>
</tr>
<tr>
<td>4/10/2006</td>
<td>ASUS_08</td>
<td>soap</td>
<td>newvolume</td>
<td>newvolume</td>
<td>4 4</td>
</tr>
</tbody>
</table>

Table I.1: Sample page of tool logging data.


**Figure I.1:** Outcome of questionnaire questions 9 and 24.

**Figure I.2:** Outcome of logged functions also covered by questionnaire questions 9 and 24.
Excellent!

-----Original Message-----
From: Rudiger Lutz
Sent: Wednesday, 20 October 2004 21:03
To: Thomas Fischer; Rudiger Lutz
Cc: Mark Burry; Peter Bowtell
Subject: AW: Spanish Project

Thomas, attached is our recent work for your information.

Rüdiger Lutz
Dear Mr. Lutz,

please find attached an initial response to your descriptions on the phone last Friday. According to your plans, the structure represents a distorted tetraherdal configuration of open sheet-metal triangles. I have put in a fair amount of visual irregularity, which can be adjusted as needed.

As I will continue working on this, I should be grateful to receive your feedback and comments.

Thanks a lot and best regards,

Thomas
From: Hugh Whitehead  
To: Thomas Fischer  
Cc: Mark Burry  
Subject: RE: Pre-, Co- and Post-Rationalisation  
Date: Mon, 5 Sep 2005 12:27:37 +0100  

Dear Thomas

Apologies for the delay in replying to your enquiry. The idea of distinguishing between three different attitudes towards rationalisation has been discussed in presentations, but has not as yet appeared in any publications. In the presentation referred to by Mark Burry the projects cited as examples were as follows:-

1) Pre-rationalised - SwissRe (30 St Mary Axe)
The concept evolved through the use of sketches and physical models and was firmly in place before parametrics were employed to refine and then implement the design. At each floor the rules are always the same but the expression is different. The dialogue between structural and cladding node geometry is dictated by rules and relationships which had already been determined at concept stage.
2) Post-rationalised - GLA (City Hall)

The competition scheme was completely reworked in order to make it deliverable as a building while retaining the original massing and energy concepts. A significant change was that the floorplates were post-rationalised to be circular so that the cladding surface could be described as a family of sheared cones, which has a natural flat panel solution. The circles are defined by their diameters which result from intersections at floor levels with the front and rear arcs of the section. This simple idea could be conveyed as a Geometry Method Statement so that all fabricators and contractors generated their own models from first principles.

3) Embedded rationale - The Sage Music Centre (Gateshead)

The project has a roof which effectively shrink-wraps three different performance spaces. The roof, which appears to be a free-form double curved surface, was required to have an economic structure which could be clad with flat panels that maximise repetition of types. The approach taken was to embed the rationale in the tools used to create the form. Parametric templates based on tangent arcs were developed for both long and cross sections and also trimming surfaces so that the form could easily be varied but the result would always be composed of torus patches. This guaranteed that the supporting ribs would always be identical and each patch could be clad with flat panels of similar size. Over a period of three months the design team used the templates to produce more than a hundred variations of roof shape as the performance spaces changed size, shape and configuration on a daily basis.

I hope this helps. The GLA project was described in some detail in the book ‘Architecture in the Digital Age (Spon Press) by Branko Kolarevic. Also enclosed is Norman’s essay ‘Design in a Digital Age’ and the bibliography for the buildings referred to.

Regards
Hugh

P.S. I could get to the .edu.hk site but could not access your .pdf which sounds intriguing

-----Original Message-----
From: Thomas Fischer
Sent: 27 August 2005 13:11
To: Hugh Whitehead
Cc: Mark Burry
Subject: Pre-, Co- and Post-Rationalisation
Dear Mr. Whitehead,

my name is Thomas Fischer, PhD student of Mark Burry at RMIT University Melbourne. I hope you could find the time to reply to my below question.

My research thesis deals with the issue of design rationalisation - a subject [for which] you as a leading expert in the area have introduced a model distinguishing between pre-, co- and post- rationalisation. I am referring to this in a paper, which I am in process of finalising for the Automation in Construction journal. The paper lacks formal references to your above-mentioned model since I have almost exclusively learned about it by word- of-mouth. Is there any publication on rationalisation that you have written, or that you are aware of, which discusses the distinction between pre-, co- and postrationalisation? Your answer would be of great help to strengthen my work.

The current draft of the paper, if you have the time and the interest, can be found here: http://people.sd.polyu.edu.hk/~sdtom/rationalising_bubble_trusses.pdf

Thanks a lot and best regards,

Thomas
This appendix presents a thought exercise that I found useful in the process of developing a relativist perspective over the course of the work presented in this thesis. The philosophical question underlying this issue is whether an objective reality exists or not, and, should it exist, whether statements about this objective reality can be justified. Förster [2003], p.293 claims that this question is in principle undecidable and relativist theory argues that some of the reasons for our inability to know the answer to this question lie in our limited means of perception and in the perspectives from which observers observe. Förster (ibid.) argues from a radical constructivist (phenomenological) viewpoint, which assumes that observed phenomena can only be studied subjectively, through human perception and consciousness. In this perspective, a main characteristic of human consciousness is that knowledge is made or “constructed” rather than
found or discovered. Accordingly, relativist investigations including those based on second-order cybernetics emphasise a first-person viewpoint. This approach is opposed to the positivist view, which assumes that knowledge can be objective and independent of an observer’s viewpoint. Knowledge is only recognised as such if it can be affirmed through strict application of formal scientific methods.

The thought exercise examines the notion of complexity, a frequently used word in architectural theory and practice. Oftentimes the word complexity seems to be used to describe attributes of something observed, such as the level of detail or the visual irregularity of an architectural structure. From a positivist viewpoint, complexity is a measure of attributes of given phenomena under examination that is independent from human perception. To measure complexity, quantitative gauges are used such as Kolmogorov complexity, which measures the length of the shortest program sufficient to produce a symbolic reproduction of the phenomenon. Phenomena whose symbolic descriptions require longer programs are viewed as having higher complexity (and hence less order). Those phenomena whose shortest describing program is as long as the phenomenon itself are viewed as having maximum complexity. In the positivist approach it is assumed that observed phenomena possess objectively assessable attributes. The following thought exercise\(^1\) demonstrates that attributes perceived to be inherent to observed phenomena are subjectively attributed by (or: subjective attributes of) the observer.

Imagine a particle system in the form of a jar of pills. The set of pills consists of bright-coloured pills and dark-coloured pills of roughly the same number. Two such sets are shown in figure L.1 below.

As illustrated on the left-hand side of figure L.1, assume a highly ordered overall

---

\(^1\)This thought exercise is inspired by Chalidze’s [2000], pp. 12 ff. explanation of irreversibility.
state with all dark pills in the bottom half of the jar and all bright pills in the upper half. When the jar is shaken, the pills are moved about in the jar and the system gradually loses its initial state of order. In other words, a program describing the initial distribution could consist of a brief instruction to place a number of dark pills into the lower half and another number of bright pills into the upper half of the jar. The program necessary to describe the less orderly distribution of colour contained in the jar after shaking would most likely need to be longer. Supporters of the positivist view would say that the system has gone from a higher degree of order to a more lower degree of order (or, that it has proceeded towards increased complexity). Now, take into account pill shape as a second particle attribute besides colour. Light and dark pills can both either be round or oval, resulting in a system that consists of four types of pills. Start again from the initial distribution shown on the left-hand side of figure L.1, in which all dark pills are at the bottom and the bright ones on top of the jar. In terms of shape of pills, however, the distribution has a lower degree of order in the positivist sense. That is, the shortest program able to
produce this distribution’s symbolic representation is as considerably longer than
assumed initially. If the jar is shaken, the pills move about and arrive at a yet
different configuration, possibly distributed somewhat as shown on the right of
figure L.1. With respect to colour, the system has again lost part of its order. But
because shape-wise the system has started out from a very low degree of order,
the new and different distribution could have increased its order. In that case,
order would have increased and decreased in the same system at the same time
through the influence of the same forces! Focusing on colour, a program to produce
a symbolic description of the jar’s content would at this later stage have to be
more extensive than the one required for describing the initial system. Focusing on
particle shape, it might however be shorter than initially required. The length of a
symbolic description-generating program that takes both phenomena into account
might not have changed very much at all.

In this thought exercise, the initial ignoring of the pills’ shape attribute appears
obvious and one might therefore be inclined to dismiss it as an avoidable mistake.
But ignorance and choices of this kind on the part of the observer are inevitably
present (consciously or not) in any observation and, as in the above thought
experiment, they can change over time. Imagine that further examination of the
pills would reveal that each pill had a small serial number printed on its back. Or
imagine that attributes at other scales such as microscopic details entered the set
of attributes taken into account by the observer. Taking these additional attributes
of the pills into account, the resulting notion of complexity (and, conversely, the
notion of order) might again change, possibly in different ways. This ambiguity of
the notion of complexity does not lie in the algorithm by which it is calculated.
It lies in the choices made in the observation preceding the calculation. This
is reflected by the practical difficulty of knowing whether the shortest possible
program for describing the phenomenon in question has already been found or whether even shorter ones are possible. The choices of attributes and algorithms used may be tightly controlled and agreed upon within groups of observers. As discussed in appendix M, this strategy can however impoverish the processes that employ it. A symbolic representation of something is by definition digital and thus, as Wilden [1980], p. 192 puts it, “full of holes” rather than “full”.

Complexity is less a measure of objective properties of observed phenomena than it is a measure of the observer’s success of explaining the phenomenon based on some chosen attributes. The less an observer succeeds to identify structuring attributes in an observed phenomenon, the more complex the phenomenon appears to the observer. An observer’s assessment of a phenomenon as “complex” may not say much about the phenomenon as long as the choice of attributes under consideration, and hence the criteria of the assessment, are unknown. This assessment can however say something about the observer’s state of confusion about the phenomenon. This is how it can be explained that architectural structures, evaluated from personal standpoints, can be assessed with varying results. The single-layer glazed space frame roof structure for the British Museum Courtyard by Foster Partners and Buro Happolt for example is described by The Guardian [2000] as “complex” while Mitchell [2005], p. 48 describes it as having “relatively low complexity”.

Förster discusses subjectivity and the role of the observer in his paper “Cybernetics of cybernetics” (see Foerster [2003], pp. 283-286). He describes the positivist position that “[t]he properties of the observer shall not enter the description of his observations” as “a peculiar delusion within our Western tradition” (ibid., p.285). Förster explains: “Consider, for instance, ‘obscenity’. There is at aperiodic intervals a ritual performed by the supreme judges of this land in which they attempt to
establish once and for all a list of all the properties that define an obscene object or act. Since obscenity is not a property residing within things (for if we show Mr. X a painting and he calls it obscene, we know a lot about Mr. X but very little about the painting), when our lawmakers will finally come up with their imaginary list we shall know a lot about them but their laws will be dangerous nonsense” (ibid., p.285).

The same goes for other kinds of properties of observable phenomena that are oftentimes alleged to reside within what is observed. Consider how the ability to play chess was once assumed to demonstrate a computer program’s high level of intelligence. This judgement has somewhat changed with the development of chess-playing software able to perform the game at master level using strategies that are likely not like those applied by human chess-players. A classic test of intelligence, Alan Turing’s [1950] so-called Turing Test, depends similarly on an observer’s judgement rather than on objective criteria. The challenge of defining creativity or good design can be seen as closely related.

In the context of design, Dorst [2004] argues that relativist (subjective or “phenomenologist”) and positivist (objective) attitudes may both be appropriate at different stages of design processes, suggesting that a relativist attitude may be more apposite in earlier, novelty-generating and ill-structured stages while a positivist attitude may be more apposite in later stages during which design ideas are implemented and presented. In his analysis of architectural representations, Goel [1995], p.180 describes a similar transition between systems of representation during architectural design processes. From discursive language as found in project briefs, non-notational early sketches are developed, which are then translated into notational systems of representation such as contract documents during the realisation phase of architectural projects.
This appendix supports the arguments presented in this thesis regarding the distinction between the analogue and the digital in the broader sense proposed by Wilden [1980] and implied by Goel [1995] (see section 1.5.2). The purpose of this appendix is to elaborate in detail on the understanding I have gained during the work presented in this thesis (see section 4.2.3) regarding the relationship of coded and uncoded knowledge and their roles in designing.

Prior to the work presented in this thesis, I was aware of references to “tacit” or implicit knowledge and to the limitations of and imposed by language. Embarking on the study reported in this thesis, however, has led me to a more profound understanding of these aspects of designing. Schön [1983], p. 51, for example, notes: “skillful action often reveals a ‘knowing more than we can say’”. Schön refers to an essay by Barnard that distinguishes “thinking processes” from “non-
logical processes”, which are “not capable of being expressed in words or as reasoning”.

I assumed that should this other kind of unspeakable knowing exist, its unspeakability would be due to some limitation in its owners’ capability of expressing it. What I did not understand is that this is rather about a quality of certain kinds of thought and knowledge than about limitations in the knower. Quite on the contrary, it seems to take sensitivity and skill on the part of the knower to be aware of this unspoken knowledge and to make use of it. Along the same lines, McKim [1980], p. 26 notes: “Computers cannot see or dream, nor can they create: computers are language-bound. Similarly, thinkers who cannot escape the structure of language, who are unaware that thinking can occur in ways having little to do with language, are often utilizing only a small part of their brain that is indeed like a computer”. This implicit, hard to express type of knowledge is characterised in this thesis as analogue, in contrast to explicit, coded digital knowledge.

The extended notions of the analogue and the digital referred to in this thesis are proposed and described by Bateson [1979], to some extent by Aicher[1991], pp. 45-52, in particular by Wilden [1980], pp. 155-201 and are implied by Goel [1995] (who uses the terms notational and non-notational but in my reading writes about the same distinction) as well as more recently by Downton [2003]. They apply more broadly than in the more commonly used technical or computational sense. The analogue, as referred to here, does not necessarily involve current fluctuations in electronic circuitry and the digital, as referred to here, does not necessarily involve

---

1Curiously, Aicher seems to mix up the relationship between the analogue and the digital as described in 4.2.3 and the relationship between guidance from inside-out and outside-in perspectives as described in 4.2.2.
computer processors or binary logic. Put simply, analogue refers to the *uncoded* and digital refers to the *coded* (see also figure 4.5). Both terms relate to *thinking and communication in general*, where, as Wilden argues, both can always be found. The transmission of a coded digital message always depends on uncoded, analogue pattern changes while the interpretation of an analogue message can result in discontinuous pattern changes. This makes their distinction observer-dependent (see Wilden [1980], pp. 166-167 and also appendix L). According to Wilden (ibid.), “The epistemological necessity of mapping discontinuity onto continuity must be emphasised. Epistemology is a matter of where you draw the line; every *logos* deals with boundaries. The same is true of any conceptual relation: metaphor and metonymy, closed and open system, energy and information - and, of course, the analog/digital line itself.

This is not to say that all knowledge is digital, although many philosophers seem to think so, or at least to behave as if it were. Most knowledge is analog. Only the divine power of abstraction (*Verstand*), to use Hegel’s term, is digital. Most of our knowledge or understanding (in the usual sense) is communicated analogically, by imitation, for example. In our universities, significantly enough, analog knowledge - and especially the (analog) context of (digital) knowledge - is generally denied, rejected, or ignored - except where its recognition can’t do much harm, as in art and music departments, or where it simply has to be taken into account, as in medical schools (which are very interested in the problem of analog simulation), for no amount of digitalization can properly describe the touch of a surgeon’s knife, which can have rather sudden either/or effects.”

The following tables list examples given by Wilden to illustrate the nature of the analogue and the digital in various contexts.
<table>
<thead>
<tr>
<th>Analog Form</th>
<th>Digital Form</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Computation</strong></td>
<td></td>
</tr>
<tr>
<td>Continuous scale</td>
<td>Discrete units (on/off)</td>
</tr>
<tr>
<td>Positive, actual quantities</td>
<td>Positive and negative representations of quantities</td>
</tr>
<tr>
<td>Quantitative plenitude</td>
<td>Logical complexity</td>
</tr>
<tr>
<td>No zero</td>
<td>Dependent on zero</td>
</tr>
<tr>
<td>No absence</td>
<td>Dependent on ‘gaps’ between elements</td>
</tr>
<tr>
<td>Always something or something else</td>
<td>All, some, nothing, or less than nothing</td>
</tr>
<tr>
<td>Units of computation may be</td>
<td>Units of computations cannot be</td>
</tr>
<tr>
<td>repeatedly divided</td>
<td>divided below the level of the discrete unit</td>
</tr>
<tr>
<td>Computation is imprecise and not</td>
<td>Precision is a function of capacity</td>
</tr>
<tr>
<td>related to capacity</td>
<td></td>
</tr>
<tr>
<td>Low signal-to-noise ratio</td>
<td>High signal-to-noise ratio^a</td>
</tr>
<tr>
<td>Concrete, necessary</td>
<td>Abstract, arbitrary</td>
</tr>
<tr>
<td>No truth functions</td>
<td>Logical calculus</td>
</tr>
</tbody>
</table>

^a Cf. von Neumann, 1951.

<table>
<thead>
<tr>
<th>Analog Aspect</th>
<th>Digital Aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1. Intraorganismic Communication</strong></td>
<td></td>
</tr>
<tr>
<td>Sequence, rhythm, frequency, spatial</td>
<td>On/off firing of neuron, logical</td>
</tr>
<tr>
<td>patterning</td>
<td>network</td>
</tr>
<tr>
<td>Memory trace (pattern)</td>
<td>Decision, recall</td>
</tr>
<tr>
<td>Total system</td>
<td>Part of system</td>
</tr>
<tr>
<td><strong>2. Interorganismic Communication</strong></td>
<td></td>
</tr>
<tr>
<td>Distinctions enabled by receiver</td>
<td>Distinctions established by sender</td>
</tr>
<tr>
<td>Context of all communication</td>
<td>Text of particular communication</td>
</tr>
<tr>
<td>Concerns relations, connections,</td>
<td>Represents limits</td>
</tr>
<tr>
<td>wholes, systems</td>
<td></td>
</tr>
<tr>
<td>Sequence and simultaneity</td>
<td>Combination</td>
</tr>
<tr>
<td>Contiguity</td>
<td>Combination</td>
</tr>
<tr>
<td>Similarity</td>
<td>Substitution</td>
</tr>
<tr>
<td><strong>3. Logical Distinctions</strong></td>
<td></td>
</tr>
<tr>
<td>Concrete</td>
<td>Abstract</td>
</tr>
<tr>
<td>Territory</td>
<td>Map</td>
</tr>
<tr>
<td>Refusal</td>
<td>absence, zero</td>
</tr>
<tr>
<td>‘More or less’</td>
<td>‘Either / or’</td>
</tr>
<tr>
<td>Difference and similarity</td>
<td>Opposition and identity</td>
</tr>
<tr>
<td>No logical typing</td>
<td>Logical typing</td>
</tr>
</tbody>
</table>
### APPENDIX M - THE DISTINCTION BETWEEN ANALOGUE AND DIGITAL

<table>
<thead>
<tr>
<th>Cannot communicate about itself</th>
<th>Communication about communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic-pragmatic</td>
<td>Syntactic</td>
</tr>
<tr>
<td>Meaning</td>
<td>Signification</td>
</tr>
<tr>
<td>Sequence and simultaneity</td>
<td>Space and time coordinates</td>
</tr>
<tr>
<td>Continuous</td>
<td>Discontinuous</td>
</tr>
<tr>
<td>Full</td>
<td>Full of holes</td>
</tr>
<tr>
<td>Whole, relations</td>
<td>Elements, entities</td>
</tr>
<tr>
<td>Maps continuums precisely</td>
<td>Can only map boundaries precisely</td>
</tr>
<tr>
<td>Presence and absence</td>
<td>Presence or absence</td>
</tr>
<tr>
<td>Similarity and contiguity</td>
<td>Code and message, substitution</td>
</tr>
<tr>
<td>‘Pre-categorial’</td>
<td>‘Categorial’</td>
</tr>
<tr>
<td>Can represent successions</td>
<td>Indicates simultaneities successively</td>
</tr>
<tr>
<td>simultaneously</td>
<td>Observer assumed to be outside</td>
</tr>
<tr>
<td>Observer in the system</td>
<td></td>
</tr>
<tr>
<td>‘Subjective’ (contextual)</td>
<td>‘Objective’</td>
</tr>
<tr>
<td>Knowledge of ‘relations’</td>
<td>Knowledge of ‘facts’</td>
</tr>
<tr>
<td>Relativistic</td>
<td>Absolutist</td>
</tr>
<tr>
<td>Ecosystems</td>
<td>Entities</td>
</tr>
<tr>
<td>Open system</td>
<td>Closure</td>
</tr>
<tr>
<td>Free flow of meaning</td>
<td>Binding of signification</td>
</tr>
<tr>
<td>‘Untamed thought’ (<em>la pensée sauvage</em>)</td>
<td>Scientific thought; rationalism, empiricism</td>
</tr>
<tr>
<td><em>Connaître</em></td>
<td><em>Savoir</em></td>
</tr>
</tbody>
</table>

### 4. Human Communication

| Senses                           | Denotative language               |
| ‘Emotion’                        | ‘Reason’                           |
| Evocation of relation            | Transmission of abstractions       |
| Presenting                       | Naming                             |
| Rich relational semantics (ambiguous) | Powerful syntax (unambiguous); weak semantics |
| Position, context, situation     | Text, message                      |
| Memory                           | Rememoration                       |
| Understandings                   | Agreements, codicils               |
| Pain is pain, pain is a sign     | ‘Pain’ is a signifier              |
| ‘Natural’ body movements         | Artificial or conventional symbols |
| Similarity and contiguity        | Metaphor and metonymy              |
| Difference, similarity           | Opposition, identity               |
| Interactive                      | Individual                         |

### 5. Language

| Refusal, repudiation, rejection, disavowal | Negation |
| Referent, goal                           | Word, means |
| Relationship                              | Concepts  |
Evocation Information
Connotation (meaning) Denotation (signification)
Command or request Report
Semantics-Pragmatics Syntactics
Present Past, present, and future
Shifters Nouns
Typography, intonation, cadence Alphabet, phonemes
loudness, frequency Poetry Prose
Evocation of images Information about concepts

6. Systems

<table>
<thead>
<tr>
<th>Relationships</th>
<th>Entities (or metaphors thereof)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use value (Real)</td>
<td>Exchange value (Symbolic or imaginary)</td>
</tr>
<tr>
<td>More-or-less</td>
<td>Either/or</td>
</tr>
<tr>
<td>Symbolic</td>
<td>Symbolic and Imaginary</td>
</tr>
<tr>
<td><strong>Aufhebung</strong></td>
<td><strong>Verneinung</strong></td>
</tr>
<tr>
<td>Process</td>
<td>Event</td>
</tr>
<tr>
<td>Quality</td>
<td>Quantity</td>
</tr>
</tbody>
</table>

b Not including language.

*General Relationship*
(In particular systems)

1. The analog is of higher logical type than the digital.
2. The digital is of higher order of organization than the analog.
3. In nature, the digital is the instrument of the analog.
4. In (western) culture, the analog is the instrument of the digital.
5. Both analog and digital systems occur in all open systems.
6. All digitalization generates paradox or oscillation at some level in the system.
7. All control processes require digital communication to set limits on positive feedback.
8. The terms ‘analog’ and ‘digital’ describe relationships in context, and not entities or ‘objective’ categories.

Dorst [2003] uses the words “Realism versus Clarity” to make essentially the same point: “[A]ny general statement (theory, model or method) about design must sacrifice some realism for the sake of clarity.” Compared to such general statements, “Case studies and anecdotes from practice can feel much more ‘real’ and appealing, and in their richness they can be of great help to designers. In such
accounts design is described holistically, nothing is left out: process, people, design problem and context of the project are all included. This makes a well written case study a pleasure to read, much livelier than any abstract theory on design will ever be. However, the drawback of case studies is that it is difficult to pinpoint what you should learn from them. People can interpret such ‘rich’ stories in any way they want, and they can always be interpreted to confirm your preconceptions.” (ibid., p. 67) Drawing on Goodman’s [1976] theory of representational systems, Goel [1995] describes three categories of symbol systems that relate to the distinction between the digital and the analogue as presented by Wilden [1980]. Figure M.1 below illustrates the three categories proposed by Goel and gives examples for each category.

Figure M.1: “Three interesting categories resulting from the theory of notationality” reproduced from Goel [1995], p. 166.

Goel’s notational systems can be described as coded, digital systems in Wilden’s [1980] sense, whereas non-notational systems in Goel’s description seem to denote what Wilden describes as uncoded, analogue systems. Goel further notes the importance of non-notational systems such as sketching in the early, exploratory stage of architectural design processes, during which much novelty is generated.
**Archimedean solids**  Thirteen convex polyhedral bodies composed of two or more types of regular polygonal faces meeting in identical vertices. See also Platonic solid.

**caltrop**  Used in this thesis to describe a configuration of four edges joined in one node point in such as way that the angle between any two of the edges is 109.47°. Named after a defence weapon placed on surfaces to stop moving attackers, which is usually made of nails or spikes in the described configuration so that one spike always points upwards.

**Cartesian**  Used in this thesis to describe spatial structuring based on straight coordinate axis that form 90° angles between them, in reference to the French mathematician and philosopher René Descartes.

**cellular automata**  Sets of abstract arrays of cells, typically simulated on a computer in one, two or tree, but possibly also higher dimensions. Each cell has a defined set of states between which it varies according to the rules associated with each state and the states of a cell’s neighbouring cells.

**design constraint**  Limitation of possible design variety.

**developable surface**  Single curved ruled surface obtained by bending or rolling a flat surface without stretching or shrinking, which can thus be flattened onto a plane without stretching, compressing or tearing. See also double curvature and ruled surface.
**dimensionality**  Attribute of constituent elements of space grids and polyhedra such as faces, edges and vertices, denoting the number of spatial dimensions into which they extend. A vertex has a dimensionality of 0 while a polygon has a dimensionality of 3.

**directrix**  Line along which a generatrix is moved to define a surface.

**double curvature**  Surface property, which states that the surface in question is curved in any two, mutually orthogonal directions. The curved part of the surface of a cylinder is single-curved while the surface of a sphere is double-curved.

**dry foam**  Liquid foam with a liquid fraction of less than 1%. See also wet foam.

**dual**  The production of a polyhedron’s dual is a transformation operation in which faces of the polyhedron are replaced by vertices and vice versa. See also truncation and stellation.

**edge**  Line connecting two vertices of a polygon or polyhedron, along which, in the case of polyhedra, faces meet. In this thesis, edges are assumed to be replaced by strut elements in the translation of space grids into space frames.

**Euclidean**  Used to describe a geometric approach based on axiomatic systems by Greek mathematician Euclid of Alexandria. Used by William Mitchel and others in the architectural design research field to describe building forms that are limited to straight, planar and rectilinear geometric arrangements.

**flat torus**  Limited area or volume, which neighbours to itself in all directions, effectively replicating in a Cartesian fashion, thus becoming virtually endless. A flat sheet, rolled into a cylinder, with the open ends of the cylinder joined to form a doughnut shape is a two-dimensional flat torus. The sheet has no edges any more and straight lines on it can extend infinitely. The same is possible in higher dimensions, for example by replicating a three-dimensional box shape in such a fashion that an exit through one of its faces is equivalent to an entry through the opposite surface.

**gas fraction**  The ratio of gas volume to total volume in a liquid-gas foam such as soap bubble foam. See also liquid fraction.

**generative design**  Design practice in which systems (typically computer software) are used as intermediate means to generate design output. While most computer software performs in highly deterministic ways, software deployed in generative design is distinguished by the degree of its non-deterministic performance, brought about by techniques such as simulated “genetic” operations, cellular automata, randomness and so forth.

**generatrix**  Line moved along a directrix to define a surface. Sometimes also referred to as “generator”.

---

346
**genotype** Term used by biologists to describe the part of an organism’s genome, which contributes to determining some characteristic of the organism that is not shared by all member of its species, such as for example blue eyes. See also phenotype.

**geodesic dome** Almost spherical, usually single-layered frame construction. The geometry is obtained by first triangulating a Platonic solid and then moving its vertices to approximate a spherical surface. First designed by Walter Bauersfeld to support a projection surface for planetarium projectors developed by the Zeiss optics company in Jena, Germany. Later further explored, patented and popularised by Buckminster Fuller.

**irregular** Uneven, unbalanced in shape or arrangement. In a more formal sense, polyhedra (and truss structures) are irregular if they contain multiple edge (strut) lengths and vertex (node) angles. In a less formal sense, polyhedra (and truss structures) are called irregular if they appear to an observer as being irregular in the more formal sense. See also regular, non-periodic.

**isotropic** invariant in all directions

**liquid fraction** The ratio of liquid volume to total volume in a liquid-gas foam such as soap bubble foam. See also gas fraction.

**parameter** Data element representing a measure that is applicable within a given context, expressed symbolically, that is, using numbers or similar coding systems.

**parametric constraint** Limitation of possible parametric variety. See also design constraint.

**patch** Used in this thesis to describe an element of a larger, composite two- or three-dimensional structure. Compositional attributes of a patch repeat throughout the larger structure, giving the larger structure somewhat uniform characteristics.

**phenotype** Term used by biologists to describe some characteristic of an organism, which is not shared by all member of its species, such as for example blue eyes. See also phenotype.

**Plateau border** Edge in liquid foam. Plateau borders are typically curved to allow foam bubbles to approximate or reach a state of minimal surface area. Plateau borders are located between three soap bubbles and therefore have a concave triangular cross-section. This cross-section shape is ignored in this thesis and edges are treated as geometrical lines.

**Platonic solid** Five convex polyhedral bodies composed of congruent regular polygonal faces with the same number of faces meeting at each vertex. See also Archimedean solid.
**polygon** Flat, two dimensional figure circumscribed by closed set of edges, which are joined at vertices. See also polyhedron.

**polyhedron** Three-dimensional body with a closed surface consisting of polygonal faces, which are joined by their edges. See also polygon.

**regular** Uniform in shape or arrangement. In a more formal sense, polyhedra (and space grids) are regular if they contain only one single edge (or strut) length and one single vertex (node) angle. In a less formal sense, polyhedra (and truss structures) are called regular if they appear to an observer as being regular in the more formal sense. See also irregular, periodic.

**ruled surface** Surface that can be described by moving a straight line through space. See also developable surface.

**stellation** Transformation operation applied to a polyhedron in which edges and faces extended until they converge in new vertices. See also dual and truncation.

**STL** Stereolithography. Rapid prototyping technology allowing machine-output of digital models. Also the name of a computer file format for this purpose.

**tetrahedral angle** The $109.47^\circ$ or $\cos^{-1} \frac{-1}{3}$ angle formed by any two lines connecting the centre of a tetrahedron to any two vertices of the tetrahedron. Four lines meeting at tetrahedral angles form a caltrop.

**tool** In this thesis, tools are understood as man-made artefacts that are external to the body and applied by somebody to tasks so as to attain somebody’s goals.

**topology** The set of qualitative characteristics of geometric figures including the number of elements such as vertices and edges and their connections, but excluding quantitative measures of distances, radii, angles and so forth. The shapes of the letters ‘O’ and ‘o’ are topologically equivalent while the shapes if the letters ‘G’ and ‘g’ (as printed here) are topologically different.

**truncation** Transformation operation applied to a polyhedron in which edges and vertices are replaced by faces. See also dual and stellation.

**vertex** Corner point at which edges meet. In this thesis, vertices are assumed to be replaced by node elements in the translation of space grids into space frames.

**vertex angle** Term I use in this thesis to describe the angle between two edges sharing a common vertex in a grid structure.

**Voronoi diagram** Partitioning of (in this thesis two- or three-dimensional) space into cells starting from a set of points within the space. Named after mathematician Georgy Voronoy.

**wet foam** Liquid foam with a liquid fraction of 1% or more. See also dry foam.