

Table of Contents

Foreword	xi
Pascal GROJEAN	
Preface	xv
Chapter 1. Why Simplicity?	1
1.1. Solving conflicting requirements	1
1.2. Three periods in IS management	5
1.2.1. Management driven by technology	5
1.2.2. Management through cost reduction	6
1.2.3. Management through value creation	8
1.3. And now ... simplicity!	10
1.3.1. Technology, cost reduction, value creation ...So what's next?	10
1.4. Plan of the book	13
Chapter 2. Complexity, Simplicity, and Abstraction	17
2.1. What does information theory tell us?	17
2.1.1. Shannon's entropy	20
2.1.2. Kolmogorov complexity	22
2.1.2.1. Complexity of objects versus complexity of binary strings.	24

2.1.2.2. Relation to Shannon’s entropy	25
2.1.2.3. Can we compute K-complexity?	26
2.1.3. Bennett’s logical depth	27
2.1.4. Abstraction in light of scale and depth	29
2.1.5. Harvesting information theory	31
2.2. What does the design tell us?	33
2.2.1. Simplicity by reduction.	36
2.2.2. Simplicity by hiding complexity.	38
2.2.2.1. Customers	40
2.2.2.2. Business analysts	40
2.2.2.3. IT personnel	41
2.2.3. Simplicity through organization	42
2.2.4. Simplicity through learning	45
2.2.4.1. Learning obviates the need to hide complexity	45
2.2.4.2. Learning allows complexity transformation.	48
2.2.5. Simplicity implies time saving.	49
2.2.5.1. Lack of time	49
2.2.5.2. How simplicity saves time	50
2.2.6. Simplicity needs trust	51
2.2.7. What does software architecture tell us?	53
2.2.7.1. The complexity of code and of IS architecture	54
2.2.8. Abstraction in software engineering.	60
2.2.8.1. Abstraction is everywhere in software.	60
2.2.8.2. Depth and scale revisited.	60
Chapter 3. Value or Values?	77
3.1. Who is concerned?	79
3.1.1. Internal stakeholders	79
3.1.2. External stakeholders	80
3.2. Concepts of value for an IS.	80
3.2.1. Book value	81
3.2.2. Net worth	82
3.2.3. Use value	83
3.2.3.1. Functional criterion	84

3.2.3.2. Non-functional criteria.	84
3.2.4. Strategic value.	86
3.2.5. Sustainability value	87
3.3. Are these values sufficient and independent? . .	90
3.3.1. IT chaos	92
3.3.2. Tech academy	92
3.3.3. Alignment trap	92
3.3.4. Users are unimportant	93
3.3.5. Business-user tyranny	93
3.3.6. Wrong direction	94
3.3.7. Architecture is a waste of money	94
3.3.8. IS heaven	95

Chapter 4. Promoting Value Through Simplicity

Chapter 4. Promoting Value Through Simplicity	97
4.1. Growing technical heterogeneity.	100
4.1.1. Openness.	104
4.1.1.1. Why complexity increases	105
4.1.1.2. Implementing simplicity	105
4.1.2. Rapid obsolescence of IT	106
4.1.2.1. Why complexity increases	110
4.1.2.2. Implementing simplicity	113
4.1.3. Absence of technological vision and leadership	118
4.1.3.1. Why complexity increases	118
4.1.3.2. Implementing simplicity	120
4.2. Changing requirements	121
4.2.1. Why complexity increases	123
4.2.2. Implementing simplicity	125
4.2.2.1. Technical answers	125
4.2.2.2. Organizational answers	129
4.3. Human factors	131
4.3.1. Multidisciplinary	132
4.3.1.1. Why complexity increases	132
4.3.1.2. Implementing simplicity	135
4.3.2. Disempowerment of IT skills	136
4.3.2.1. Why complexity increases	136

4.3.2.2. Implementing simplicity	140
4.3.3. Local interest is not global interest	143
4.3.3.1. Why complexity increases	143
4.3.3.2. Implementing simplicity	145
Chapter 5. Simplicity Best Practices	149
5.1. Putting simplicity principles into practice	149
5.2. Defining a generic IS	149
5.3. A simplicity framework	152
5.3.1. Simplicity in hardware	153
5.3.1.1. Growing technical heterogeneity	153
5.3.1.2. Changing requirements	155
5.3.1.3. Human factors	155
5.3.2. Simplicity in software – data access	156
5.3.2.1. Growing technical heterogeneity	156
5.3.2.2. Changing requirements	158
5.3.2.3. Human factors	159
5.3.3. Simplicity in software – services	160
5.3.3.1. Growing technical heterogeneity	161
5.3.3.2. Changing requirements	162
5.3.3.3. Human factors	163
5.3.4. Simplicity in software–user interface	165
5.3.4.1. Growing technical heterogeneity	165
5.3.4.2. Changing requirements	166
5.3.4.3. Human factors	167
5.3.5. Simplicity in Functional Architecture	169
5.3.5.1. Growing technical heterogeneity	169
5.3.5.2. Changing requirements	169
5.3.5.3. Human factors	170
Conclusion	173
APPENDICES	177
Appendix 1. Digging into Information Theory	179
A1.1. Shannon entropy	179
A1.2. Shannon entropy in short	182

A1.3. Kolmogorov complexity	183
A1.4. Choosing a scale of description	186
A1.5. Relation to Shannon entropy	187
A1.6. Computing the Kolmogorov complexity	187
A1.7. Kolmogorov complexity in short	189
A1.8. Bennett's logical depth.	189
A1.9. Bennett's logical depth in short	192
Appendix 2. Two Measures of Code Complexity . .	195
A2.1. Cyclomatic complexity	195
A2.2. An example of a scale-invariant complexity measure.	198
A2.3. Conclusion	204
Appendix 3. Why Has SOA Failed So Often?	207
A.3.1. The need for flexibility	207
A.3.2. First issue: no suitable enterprise architecture	208
A.3.3. Second issue: no data integration.	209
A.3.4. Identifying the operating model.	210
A.3.4.1. Data integration	211
A.3.4.2. Process standardization.	212
A.3.5. Which models are compatible with SOA?	213
A.3.5.1. Diversification model.	214
A.3.5.2. Replication model	215
A.3.5.3. Coordination model	215
A.3.5.4. Unification model	216
A.3.6. Conclusion on SOA.	217
Bibliography	219
Index	221