THIRD EDITION

THE ART OF SYSTEMS ARCHITECTING

MARK W. MAIER
EBERHARDT RECHTIN

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

Appendix A: Heuristics for Systems-Level Architecting

Experience is the hardest kind of teacher.
It gives you the test first and the lesson afterward.

Susan Ruth, 1993

Introduction: Organizing the List

The heuristics to follow were selected from Rechtin 1991,[1] the *Collection of Student Heuristics in Systems Architecting, 1988–1993*,[2] and from subsequent studies in accordance with the selection criteria of Chapter 2. The list is intended as a tool store for top-level systems architecting. Heuristics continue to be developed and refined not only for this level, but for domain-specific applications as well, often migrating from domain-specific to system level and vice versa.*

* The manufacturing, social, communication, software, management, business, and economics fields are particularly active in proposing and generating heuristics — though they usually are called principles, laws, rules, or axioms.

For easy search and use, the heuristics are grouped by architectural task and categorized by being either descriptive or prescriptive — that is, by whether they describe an encountered situation or prescribe an architectural approach to it, respectively.

There are over 180 heuristics in the listing to follow, far too many to study at any one time. Nor were they intended to be. The listing is intended to be scanned as one would scan software tools on software store shelves, looking for ones that can be useful immediately, but remembering that others are also there. Although some are variations of other heuristics, the vast majority stand on their own, related primarily to others in the near vicinity on the list. Odds are that the reader will find the most interesting heuristics in clusters, the location of which will depend on the reader's interests at the time. The section headings are by architecting task. A "D" signifies a descriptive heuristic; a "P" signifies a prescriptive one. When readily apparent, prescriptions are grouped by insetting under appropriate descriptions or alternate prescriptions; otherwise not. In the interests of brevity, an individual heuristic is listed in the task where it is most likely to be used most often. As noted in Chapter 2, some 20% can be tied to related ones in other tasks.

A major difference between a heuristic and an unsupported assertion is the credibility of the source. To the extent possible, the heuristics are credited to the individuals who, to the authors' knowledge, first suggested them. To further aid the reader in judging credibility or in finding the sources, the heuristics to follow are given symbols. These symbols indicate the following:

[ ] *An informal discussion* with the individual indicated, unpublished.

( ) A formal, dated source, with examples, located in the University of Southern California (USC) Master of Science in Systems Architecture and Engineering (MS-SAE) program archive, especially in the Collection of Student Heuristics in Systems Architecting, 1988–1993. For further information, contact the Master of Science Program in Systems Architecture and Engineering, USC School of Engineering, University Park, Los Angeles, California 90089-1450.

* Rechtin 1991, where it is sourced more formally. By permission of Prentice Hall Inc., Englewood Cliffs, New Jersey 07632.

Bold Key words useful for quick search. Otherwise, heuristics to follow are in plain type to make page reading easier. Real-world examples of each can be found in the references indicated.

The authors apologize in advance for any miscrediting of sources. Corrections are welcome. The readers are reminded that not all heuristics apply to all circumstances, just most to most.

Heuristic Tool List

Multitask Heuristics

D **Performance, cost, and sche**
At least one of the three mu

D With few exceptions, sched
cost **overruns** will **not**, and

D The **time to completion** is p
to the time planned to dat
time to go.

D **Relationships** among the
added value.*

D **Efficiency** is inversely pr
King 1992)

D **Murphy's Law**, "If anythin

P **Simplify**. Simplify. Sim

P The first line of defense

P Simplify, combine and

P Simplify with smarter

P The most **reliable part**
because it isn't needed

D One person's **architectur**
system is another's compo

P In order to **understan**
stand everything. (Ari

P Don't confuse the fun
the system. (Jerry Oliv

D In general, each **system le**
(G. G. Lendaris 1986)

P Leave the **specialties**
by the architect is onl
critical to the system
architect must have **a**
about its criticality an

P Complex systems w
architecture much m
forms than if there a

D Particularly for social s
that count.

* As indicated in the introduction to
is taken from Rechtin 1991. (With
Jersey.)

Heuristic Tool List

Multitask Heuristics

D **Performance, cost, and schedule** cannot be specified independently. At least one of the three must depend on the others.**

D With few exceptions, schedule **delays** will be **accepted** grudgingly; cost **overruns** will **not**, and for good reason.

D The **time to completion** is proportional to the ratio of the time spent to the time planned to date. The greater the ratio is, the longer the time to go.

D **Relationships** among the elements are what give systems their added value.*

D **Efficiency** is inversely proportional to **universality**. (Douglas R. King 1992)

D **Murphy's Law**, "If anything can go wrong, it will."*

P **Simplify**. Simplify. Simplify.*

P The first line of defense against complexity is simplicity of design.

P Simplify, combine and eliminate. (Suzaki 1987)

P Simplify with smarter elements. (N. P. Geiss 1991)

P The most **reliable part** on an airplane is the one that isn't there — because it isn't needed. [DC-9 Chief Engineer 1989]

D One person's **architecture** is another person's **detail**. One person's system is another's component. [Robert Spinrad 1989]*

P In order to **understand anything**, you must not try to understand everything. (Aristotle, 4th century B.C.)

P Don't confuse the functioning of the parts for the functioning of the system. (Jerry Olivieri 1992)

D In general, each **system level** provides a context for the level(s) below. (G. G. Lendaris 1986)

P Leave the **specialties** to the specialist. The level of detail required by the architect is only to the depth of an element or component critical to the system as a whole. (Robert Spinrad 1990) But the architect must have **access** to that level and know, or be informed, about its criticality and status. (Rechtin 1990)

P Complex systems will develop and **evolve** within an overall architecture much more rapidly if there are **stable intermediate** forms than if there are not. (Simon 1969)*

D Particularly for social systems, it's the **perceptions**, not the facts, that count.

* As indicated in the introduction to this appendix, an asterisk indicates that this heuristic is taken from Rechtin 1991. (With permission of Prentice Hall, Englewood Cliffs, New Jersey.)

D In introducing technological and **social** change, **how** you do it is often more important than **what** you do.*

  P If social cooperation is required, the **way** in which a system is **implemented** and introduced must be an **integral part** of its architecture.*

D If the **politics** don't fly, the hardware never will. (Brenda Forman 1990)

  D Politics, not technology, sets the limits of what technology is allowed to achieve.

  D Cost rules.

  D A strong, coherent constituency is essential.

  D Technical problems become political problems.

  D There is no such thing as a purely technical problem.

  D The best engineering solutions are not necessarily the best political solutions.

D **Good products** are not enough. Implementations matter. (Morris and Ferguson 1993)

  P To remain competitive, determine and **control the keys** to the architecture from the very beginning.

Scoping and Planning

> The beginning is the most important part of the work.
>
> **Plato, 4th century B.C.**

> **Scope!** Scope! Scope!
>
> **William C. Burkett, 1992**

D **Success** is defined by the **beholder**, not by the architect.*

  P The most important single element of success is to **listen** closely to what the customer perceives as his requirements and to have the will and ability to be responsive. (J. E. Steiner 1978)*

  P **Ask early** about how you will **evaluate** the success of your efforts. (F. Hayes-Roth et al., 1983)

  P For a system to meet its **acceptance criteria** to the satisfaction of all parties, it must be architected, designed, and built to do so — no more and no less.*

  P Define how an **acceptance** criterion is to be certified at the same time the criterion is established.*

  D Given a **successful** organization or system with valid criteria for success, there are some things it **cannot do** — or at least not do well. Don't force it!

d **social** change, **how** you do it is
you do.*
red, the **way** in which a system is
d must be an **integral part** of its
re never will. (Brenda Forman 1990)
the limits of what technology is
cy is essential.
olitical problems.
rely technical problem.
s are not necessarily the best politi-
Implementations matter. (Morris
mine and **control the keys** to the
ginning.
important part of the
Plato, 4th century B.C.
lliam C. Burkett, 1992
, not by the architect.*
ment of success is to **listen** closely
s as his requirements and to have
nsive. (J. E. Steiner 1978)*
ll **evaluate** the success of your
83)
ance criteria to the satisfaction of
d, designed, and built to do so —
rion is to be certified at the same
.*
n or system with valid criteria for
it **cannot do** — or at least not do

P The **strengths** of an organization or system in one context can be its **weaknesses** in another. Know when and where!*

D There's nothing like being the **first success**.*

P If at first you don't succeed, but the architecture is sound, try, try again. Success sometimes is where you find it. Sometimes it finds you.*

D A system is successful when the **natural intersection** of technology, politics, and economics is found. (A. D. Wheelon 1986)*

D Four questions, **the Four Who's**, need to be answered as a self-consistent set if a system is to succeed economically; namely, who benefits? who pays? and, as appropriate, who loses? who provides

D **Risk** is (also) defined by the **beholder**, not the architect.

P If being absolute is impossible in estimating system risks, then be relative.*

D **No** complex system can be **optimum** to all parties concerned, nor all functions optimized.*

P Look out for hidden agendas.*

P It is sometimes more important to know **who** the customer is than to know what the customer **wants**. (Whankuk Je 1993)

D The phrase, **"I hate it,"** is direction. (Lori I. Gradous 1993)

P Sometimes, but not always, the best way to solve a difficult problem is to **expand** the problem, itself.*

P Moving to a **larger purpose** widens the range of solutions. (Gerald Nadler 1990)

P Sometimes it is necessary to **expand the *concept*** in order to simplify the problem. (Michael Forte 1993)

P [If in difficulty,] **reformulate** the problem and re-allocate the system functions. (Norman P. Geis 1991)

P Use **open** architectures. You will need them once the market starts to respond.

P Plan to **throw one away**. You will anyway. (F. P. Brooks, Jr. 1982)

P You can't avoid **redesign**. It's a natural part of design.*

P Don't make an architecture **too smart** for its own good.*

D Amid a **wash of paper**, a small number of documents become critical pivots around which every project's management revolves. (F. P. Brooks, Jr. 1982)*

P Just because it's written, doesn't make it so. (Susan Ruth 1993)

D In architecting a new [software] program, all the **serious mistakes** are made in the **first day**. [Spinrad 1988]

P The most **dangerous** assumptions are the **unstated** ones. (Douglas R. King 1991)

D Some of the **worst** failures are **systems** failures.

D In architecting a new [aerospace] system, by the time of the first **design review**, performance, cost, and schedule have been **predetermined**.

One might not know what they are yet, but to first order all the critical assumptions and choices have been made which will determine those key parameters.*

P **Don't assume** that the original statement of the problem is necessarily the best, or even the right, one.*

P **Extreme** requirements, expectations, and predictions should remain under challenge. throughout system design, implementation, and operation.

P Any extreme requirement must be intrinsic to the system's design philosophy and must validate its selection. "Everything must pay its way on to the airplane." [Harry Hillaker 1993]

P **Don't assume** that previous **studies** are necessarily complete, current, or even correct. (James Kaplan 1992)

P Challenge the process and solution, for surely someone else will do so. (Kenneth L. Cureton 1991)

P Just because it worked in the past there's no guarantee that it will work now or in the future. (Kenneth L. Cureton 1991)

P Explore the situation from more than one point of view. A seemingly impossible situation might suddenly become transparently simple. (Christopher Abts 1988)

P **Work forward and backward.** (A set of heuristics from Rubinstein 1975)*

Generalize or specialize.
Explore multiple directions based on partial evidence.
Form stable substructures.
Use analogies and metaphors.
Follow your emotions.

P Try to hit a solution that, at worst, won't put you **out of business.** (Bill Butterworth as reported by Laura Noel 1991)

P The **order** in which decisions are made can change the architecture as much as the decisions themselves. (Rechtin 1975, IEEE SPECTRUM)

P Build in and maintain **options** as long as possible in the design and build of complex systems. You will need them. OR … Hang on to the agony of decision as long as possible. [Robert Spinrad 1988]*

P Successful architectures are **proprietary, but open.** [Morrison and Ferguson 1993]

D Once the architecture begins to take shape, the sooner contextual constraints and **sanity checks** are made on assumptions and requirements, the better.*

D Concept **formulation** is complete when the **builder** thinks the system can be built to the **client's** satisfaction.*

D The **realities** at the end of the conceptual phase are not the models but the **acceptance criteria.***

P Do the **hard parts** first.

P Firm **commitments** are b

*Modeling**

P If you can't analyze it, don't
D Modeling is a **craft** and at ti
D A **vision** is an **imaginary a**
the rest of the models. (M. B
D From psychology: If the con
different from those in the
model of the topic and no c
telecommunications: The **b**
nal model of the transmitte
Lehan 1954]*
D A model is not **reality.***
D The **map** is not the terri
P Build **reality checks** i
Dumas 1989]*
P Don't believe **nth order**
[R. W. Jensen circa 1989
D Constants aren't and variab
D One **insight** is worth a thou
P Any war game, system
easily be explained on
less, it is probably dang
D Users develop **mental mod**
user-to-system interface. (J
D If you can't explain it in **fi**
or it doesn't work. (Darcy
P The **eye** is a fine **architect.**
D A good solution somehow
P **Taste:** an aesthetic fe
only when no more di
[Robert Spinrad 1994]
P Regarding **intuition,** t

Prioritizing (Trades, Options

D In any resource-limited si
product is determined by
P When choices must be ma
tion, **choose** the best avai

* See also Chapters 3 and 4.

P Firm **commitments** are best made after the **prototype works**.

*Modeling**

P If you can't analyze it, don't build it.
D Modeling is a **craft** and at times an art. (William C. Burkett 1994)
D A **vision** is an **imaginary architecture …** no better, no worse than the rest of the models. (M. B. Renton Spring 1995)
D From psychology: If the concepts in the mind of one person are very different from those in the mind of the other, there is no **common model** of the topic and no communication. [Taylor 1975] OR … From telecommunications: The **best receiver** is one that contains an internal model of the transmitter and the channel. [Robert Parks & Frank Lehan 1954]*
D A model is not **reality**.*
- D The **map** is not the territory. (Douglas R. King 1991)*
- P Build **reality checks** into model-driven development. [Larry Dumas 1989]*
- P Don't believe **nth order consequences** of a first order [cost] model. [R. W. Jensen circa 1989]

D Constants aren't and variables don't. (William C. Burkett 1992)
D One **insight** is worth a thousand **analyses**. (Charles W. Sooter 1993)
P Any war game, systems analysis, or study whose results can't easily be explained on the back of an envelope is not just worthless, it is probably dangerous. [Brookner-Fowler circa 1988]
D Users develop **mental models** of systems based [primarily] upon the user-to-system interface. (Jeffrey H. Schmidt)
D If you can't explain it in **five minutes**, either you don't understand it or it doesn't work. (Darcy McGinn 1992 from David Jones)
P The **eye** is a fine **architect**. Believe it. [Wernher von Braun 1950]
D A good solution somehow **looks nice**. (Robert Spinrad 1991)
- P **Taste:** an aesthetic feeling that will accept a solution as right only when no more direct or simple approach can be envisaged. [Robert Spinrad 1994]
- P Regarding **intuition**, trust but **verify**. (Jonathan Losk 1989)

Prioritizing (Trades, Options, and Choices)

D In any resource-limited situation, the **true value** of a given service or product is determined by what one is willing to **give up** to obtain it.
P When choices must be made with unavoidably inadequate information, **choose** the best available and then **watch** to see whether future

* See also Chapters 3 and 4.

solutions appear faster than future problems. If so, the choice was at least adequate. If not, go back and **choose** again.*

P When a decision makes sense through several different **frames,** it's probably a good decision. (J. E. Russo 1989)

D The **choice** between architectures may well depend upon which set of **drawbacks** the client can handle best.*

P If trade results are inconclusive, then the wrong selection criteria were used. Find out [again] what the customer wants and why they want it, then repeat the trade using those factors as the [new] selection criteria. (Kenneth Cureton 1991)

P The triage: Let the dying die. Ignore those who will recover on their own. And treat only those who would die without help.*

P Every once in a while you have to go back and see what the real world is telling you. [Harry Hillaker 1993]

Aggregating ("Chunking")

P **Group** elements that are strongly **related** to each other, **separate** elements that are unrelated.

D Many of the **requirements** can be **brought together** to complement each other in the total design solution. Obviously the more the design is put together in this manner, the more probable the overall success. (J. E. Steiner 1978)

P Subsystem interfaces should be drawn so that each subsystem can be implemented independently of the specific implementation of the subsystems to which it interfaces. (Mark Maier 1988)

P Choose a configuration with **minimal communications** between the subsystems. (computer networks)*

- P Choose the elements so that they are as independent as possible; that is, elements with low external complexity (low coupling) and high internal complexity (high cohesion). (Christopher Alexander 1964 modified by Jeff Gold 1991)*
- P Choose a configuration in which local activity is high speed and global activity is slow change. (P. J. Courtois 1985) *

P Poor aggregation results in **gray** boundaries and **red** performance. (M. B. Renton Spring 1995)

- P **Never aggregate** systems that have a **conflict** of interest; partition them to ensure checks and balances. (Aubrey Bout 1993)
- P **Aggregate** around **"testable"** subunits of the product; **partition** around logical **subassemblies.** (Ray Cavola 1993)
- P Iterate the partition/aggregation procedure until a model consisting of **7 ± 2 chunks** emerge. (Moshe F. Rubinstein 1975)

P The **optimum number** of architectural elements is the amount that leads to distinct **action**, not general planning. (M. B. Renton Spring 1995)

P System structure should resemble functional structure.*

P Except for good and sufficient reasons, **functional and physical** structuring should match.*

P The architecture of a **support** element must **fit** that of the system which it supports. It is easier to match a support system to the human it supports than the reverse.*

P **Unbounded limits** on element behavior may be a **trap** in unexpected scenarios. [Bernard Kuchta 1989]*

Partitioning (Decompositioning)

P Do not **slice** through regions where **high rates** of information exchange are required. (computer design)*

D The greatest **leverage** in architecting is at the **interfaces**.*

P **Guidelines** for a good quality interface specification: They must be simple, unambiguous, complete, concise, and focus on substance. Working documents should be the same as customer deliverables; that is, always use the customer's language, not engineering jargon. [Harry Hillaker 1993]

P The **efficient architect**, using contextual sense, continually looks for likely **misfits** and redesigns the architecture so as to eliminate or minimize them. (Christopher Alexander 1964)* It is inadequate to architect up to the boundaries or **interfaces** of a system; one must architect **across** them. (Robert Spinrad as reported by Susan Ruth 1993)

P Since boundaries are inherently limiting, look for solutions outside the boundaries. (Steven Wolf 1992)

P Be prepared for **reality** to add a few interfaces of its own.*

P Design the structure with **good "bones."***

P Organize personnel tasks to **minimize** the **time** individuals spend interfacing. (R. C. Tausworthe 1988)*

Integrating

D **Relationships** among the elements are what give systems their **added value**.*

P The greatest leverage in system architecting is at the interfaces.*

P The greatest **dangers** are also at the interfaces. [Raymond 1988]

P Be sure to ask the question, "What is the worst thing that other elements could do to you across the interface?" [Kuchta 1989]

D Just as a piece and its template must match, so must a system and the resources which make, test, and operate it. Or, more briefly, the **product and process** must match. Or, by extension, a system architecture cannot be considered complete lacking a suitable match with the process architecture.*

P When confronted with a particularly difficult interface, try changing its **characterization.***

P Contain **excess energy** as close to the source as possible.*

P Place barriers in the paths between energy sources and the elements the energy can damage. (Kjos 1988)*

Certifying (System Integrity, Quality, and Vision)

D As **time to delivery** decreases, the **threat** to functionality increases. (Steven Wolf 1992)

P If it is a good design, **insure** that it stays **sold**. (Dianna Sammons 1991)

D Regardless of what has gone before, the **acceptance criteria** determine what is actually built.*

D The number of **defects remaining** in a (software) system after a given level of test or review (design review, unit test, system test, etc.) is proportional to the **number found** during that test or review.

P **Tally** the defects, analyze them, **trace** them to the source, make corrections, keep a **record** of what happens afterwards and keep **repeating** it. [Deming]

P **Discipline.** Discipline. Discipline. (Douglas R. King 1991)

P The principles of **minimum communications** and proper partitioning are key to system testability and **fault isolation**. (Daniel Ley 1991)*

P **The five whys** of Toyota's lean manufacturing. (To find the basic cause of a defect, keep asking "why" from effect to cause to cause five times.)

D The test **setup** for a system is itself a system.*

P The test system should always allow a part to pass or fail on its own merit. [James Liston 1991]*

P To be tested, a system must be designed to be tested.*

D An element **"good enough"** in a small system is unlikely to be good enough in a more complex one.*

D Within the same class of products and processes, the **failure rate** of a product is linearly proportional to its **cost.***

D The **cost** to find and **fix** an inadequate or failed part increases by an **order of magnitude** as it is successively incorporated into higher levels in the system.

P The least expensive and most effective place to find and fix a problem is at its source.

D Knowing a **failure has occurred** is more important than the actual failure. (Kjos 1988)

D **Mistakes** are **understandable**, failing to report them is **inexcusable.**

D Recovery from failure or flaw is not complete until a specific mechanism, **and no other**, has been shown to be the cause.*

D Reducing **failure rate** by each **factor of two** takes as much effort as the original development.*

D **Quality** can't be tested in, it has to be **built in.***

- D You can't achieve quality … unless you specify it. (Deutsch 1988)
- P Verify the quality close to the source. (Jim Burruss 1993)
- P The five why's of Japan's lean manufacturing. (Hayes et al. 1988)[3]
- D High-**quality**, reliable systems are produced by high-quality architecting, engineering, design and manufacture, **not by inspection,** test, and rework.*
- P Everyone in the development and production line is both a customer and a supplier.

D Next to interfaces, the greatest **leverage** in architecting is in aiding the recovery from, or exploitation of, **deviations** in system performance, cost or schedule.*

Assessing Performance, Cost, Schedule, and Risk

D A good design has benefits in more than one area. (Trudy Benjamin 1993)

D System quality is defined in terms of customer satisfaction, not requirements satisfaction. (Jeffrey Schmidt 1993)

D If you think your **design** is perfect, it's only because you haven't shown it to **someone else**. [Harry Hillaker, 1993]

- P Before proceeding too far, **pause and reflect!** Cool off periodically and seek an independent review. (Douglas R. King 1991)

D Qualification and **acceptance** tests must be both definitive and **passable.***

- P High **confidence**, not test completion, is the **goal** of successful qualification. (Daniel Gaudet 1991)
- P Before ordering a **test** decide what you will do if it is 1) **positive** or if 2) it is negative. If both answers are the same, **don't do** the test. (R. Matz, M.D. 1977)

D **"Proven"** and **"state of the art"** are mutually **exclusive** qualities. (Lori I. Gradous 1993)

D The **bitterness** of **poor performance** remains long after the sweetness of low prices and prompt delivery are forgotten. (Jerry Lim 1994)

D The **reverse of diagnostic** techniques are good architectures. (M. B. Renton 1995)

D Unless everyone who **needs to know** does know, somebody, somewhere will foul up.

P Because there's no such thing as immaculate communication, don't ever stop **talking** about the system. (Losk 1989)*

D Before it's tried, it's **opinion**. After it's tried, it's **obvious**. (Wm. C. Burkett 1992)

D Before the **war**, it's opinion. After the war, it's too late! (Anthony Cerveny 1991)

D The first **quick look** analyses are often **wrong**.*

D In correcting system deviations and failures, it is important that all the participants know not only **what** happened and how it happened, but **why** as well.*

P Failure reporting without a **close out** system is meaningless. (April Gillam 1989)

P Common, if undesirable, responses to **indeterminate outcomes** or failures:*

If it **ain't broke**, don't fix it.
Let's **wait and see** if it goes away or happens again.
It was just a **random** failure. One of those things.
Just treat the **symptom**. Worry about the cause later.
Fix everything that might have caused the problem.
Your **guess** is as good as mine.

D Chances for recovery from a **single failure** or flaw, even with complex consequences, are fairly good. Recovery from **two or more** independent failures is unlikely in real time and uncertain in any case.*

Re-Architecting, Evolving, Modifying, and Adapting

The test of a good architecture is that it will last.
The sound architecture is an enduring pattern.

[Robert Spinrad 1988]

P The team that created and built a presently successful product is often the best one for its evolution — but seldom for creating its replacement.

D If you **don't understand** the existing system, you can't be sure you're rearchitecting a **better** one. (Susan Ruth 1993)

P When implementing a change, keep some elements constant to provide an **anchor** point for people to cling to. (Jeffrey H. Schmidt 1993)

P In large, mature systems, **evolution** should be a process of **ingress** and **egress**. (IEEE 1992, Jeffrey Schmidt 1992)

- P Before the change, it is your opinion. After the change it is your problem. (Jeffrey Schmidt 1992)
- D Unless constrained, **rearchitecting** has a natural tendency to proceed unchecked until it results in a substantial transformation of the system. (Charles W. Sooter 1993)
- D Given a change, if the anticipated actions don't occur, then there is probably an invisible barrier to be identified and overcome. (Susan Ruth 1993)

Exercises

Exercise: What favorite heuristics, rules of thumb, facts of life, or just plain common sense do you apply to your own day-to-day living — at work, at home, at play? What heuristics have you heard on TV or the radio (for example, on talk radio, action TV, children's programs)? Which ones would you trust?

Exercise: Choose a system, product, or process with which you are familiar and assess it using the appropriate foregoing heuristics. What was the result? Which heuristics are or were particularly applicable? What further heuristics were suggested by the system chosen?

Were any of the heuristics clearly incorrect for this system? If so, why?

Exercise: Try to spot heuristics and insights in the technical literature. Some are easy; they are often listed as principles or rules. The more difficult ones are buried in the text but contain the essence of the article or state something of far broader application than the subject of the piece.

Exercise: Try to create a heuristic of your own — a guide to action, decision making, or to instruction of others.

Notes and References

1. Rechtin, E., *Systems Architecting, Creating and Building Complex Systems*. Englewood Cliffs, NJ: Prentice Hall, 1991. Note that throughout chapter, this reference will be referred to as Rechtin 1991.