2. The failure of design methods

2.1 The general failure of design methods

Having said this much about design methods, there is but one thing to add: They don't work, and they don't work at all. In spite of all the good motives—the need for potent and up-to-date design procedures, the noble cause of being rational, and so on—the failure of these methods is a very solid and widely recognized fact, as is the thoroughness of this failure.

A number of circumstances bear testimony to this. One is that it was the original advocates of these methods who documented the failure, and who then abandoned them altogether. This is a unique circumstance; usually, it takes critics from the outside to bring a failure into the light, and this is typically met with a frenetic defense from the original proponents who rarely change sides, either refusing to see or accept the failure. Here, the pioneers allowed themselves to recognize the failure of their own ideas and inventions, and to publicly state this as a fact.

A second circumstance is how exceptionally soon this reversal came. Alexander's classical description of his method was published in 1964. In 1966 he wrote an essay explaining why it didn't work. When he was interviewed about the state of the field in 1971, he dismissed design methods completely:

> And there is so little in what is called "design methods" that has anything useful to say about how to design buildings that I never even read the literature any more.
>
> ... I think I just have to be consistent here. 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 creating buildings and actually trying to create buildings, I wouldn't give a penny for their efforts.

Also Jones (1970) acknowledged the problems with these methods, and the lack of success stories. This he did already in the original edition; even in the same paragraph as where he stated the *need* for these methods:

> However, it is not obvious that the new methods that are reviewed in this book are any better. *There is not much evidence that they have been used with success, even by their inventors* ... The usual difficulty is that of losing control of the design situation once one is committed to a systematic procedure which seems to fit the problem less and less as designing proceeds. (p. 27, my italics)

Later, Jones too would reject them completely (cf. later editions of Jones 1970). The lack of successful applications mentioned here is a third aggravating circumstance which was also generally recognized; Rittel was another pioneer who acknowledged this absence:

> Q: What kinds of problems has design methodology successfully attacked?
>
> ... If you are asking for examples from architectural design I wouldn't know of any building that has been done discernibly better than buildings done in the conventional way.
>
> ... I would say that the corporations or other planning institutions who seriously tried to accomplish something with the [design] methods have been disappointed, and that there is a considerable "hangover" from these methods. (Rittel 1972)

Also Alexander and Broadbent made similar comments:

> In short, my feeling about methodology is that there are certain mundane problems which it has solved—and I mean really incredibly mundane. (Alexander 1971)

> Yet asked to catalogue its achievements, in terms of buildings built, cities designed, and so on, most of its advocates find themselves in difficulties. (Broadbent 1979, p. 41)

Even the *attempts* to use the methods have been exceptionally rare. According to Lawson (1980), there has been only one documented attempt to use Alexander's method (Hanson 1969).

Since this failure is very well documented, I will not belabor the argument here; the existing critiques are quite sufficient. (Beyond the sources that have been cited here, see also e.g. Broadbent 1973, Lawson 1980, and several papers in Cross 1984).

2.2 The classical mistake

When it comes to understanding why the methods failed, the fundamental problems can also be traced all the way back to Pappus. The problems that are present already in his original account have since been carried over into all methods that descend from it, including the most modern ones. I will here begin by briefly examining these problems in the original context of geometry proofs.

The structure of the genuine process

The difference between analysis and synthesis does not stand out as clearly in a proof as it does in a conventional mathematical problem, the other of the two kinds of use that Pappus describes. In solving a problem, analysis consists in figuring out what calculatons are needed for reaching the answer, and synthesis of actually carrying out these calculations. As a consequence, no calculations are to be made during the analysis; this phase serves only to determine which ones *should be* performed (cf. Polya).

This division is however not made in practice, as anyone can tell from their own experience: When working on a mathematical problem, you do not refrain from attempting to calculate the answer as you are trying to find the solution. Usually, you do quite the opposite: you carry out various tentative calculations in order to figure out how to reach the solution.

You do this for several reasons: For instance, to see what you can get from what is given; you can then use this for further calculations. Or, when you have an idea for a solution, or for a part of it, to evaluate what you have and to see where it leads. Many of these calculations are dead ends; still, one or two may give you the critical insights. Why would you not make these interim calculations? There is no good reason not to, so people do make them.

Therefore, there is no phase of pure analysis, as this activity is intermingled with synthesis; nor is analysis alone sufficient to produce a solution.

As a consequence, when you have figured out how to solve the problem, then this is hardly followed by a *bona fide* synthesis phase. At that point you have already performed most or all of the necessary calculations, in particular when testing to see whether you indeed have reached the correct solution. Thus, there is hardly such thing as a synthesis phase that *a*) is performed after the solution has been found, *b*) is separate from an analysis phase, *c*) that follows after analysis, or *d*) that even exists at all.

But the work that *does* remain when you have reached the solution, especially if you are to present it to someone else, is to go back to the pieces of writing on your paper, extract the parts that go into the solution, and write them down in the logical order. *This* is how you obtain the straight path from what is given to the solution, the sequence of steps that corresponds to a proof, and that according to Pappus is followed during analysis as well as synthesis, without redundant parts and dead ends.

In actuality, this path is never followed; not during any part of the process. Hence it is a mistake to see this as plan following, or to think of the proof as a plan. It cannot even be *known* until at the very end of the process, when you also have the answer. Only then can you extract this path, and it still requires effort to obtain it: you have to look back at what you have done, go through your scribbles, and then assemble the pieces into a tidy, linear sequence.

Instead there is just one process, where the functions of analysis and synthesis are two aspects of the same activity, not two different activities, stages or processes. The structure of this joint process is anything but a clean progression from desired result to given facts, neither backward nor forward. This process will by necessity follow many routes that turn out to be dead ends, and others that are not used in the final proof. (All these things given that the proof sought or problem to be solved is not trivial; in this case the problem is so simple that it is possible to go "straight" to the correct solution.)

Pappus' method describes the product, not the process

So Pappus' method can neither be used to solve problems as he claims, nor can it be a reflection of how Euclid or other geometers actually worked. Since this mistake seems so obvious, then why was it made, and where did it come from? The basic mistake that Pappus made was to conflate the structures of the resulting proof, and that of the process that produced it. Most likely, this was because the method was established not from observing the actual *work* behind the proof, but only the *result* of this work—that is, the proof itself.

This can be seen by comparing Pappus' method with the structure of the geometrical proofs. The example in the figure is one of Pappus' own proofs, taken from Hintikka & Remes (1974, pp. 22f). Their account consists of Pappus' full original text with annotations; for brevity I have included only what is relevant to the present argument. Pappus' original text is in italics, and the numbered headings in small caps (2. ANALYSIS, etc.), which correspond to the elements

of his method, were inserted by Hintikka & Remes. In particular, note how the steps of the analysis are repeated backward in the synthesis (cf. figure 1.5):

Hintikka's annotations:	*Pappus' original text:*
1. ENUNCIATION	
1A. THAT WHICH IS GIVEN	*Let* ABG *be a circle with the centre* E. *Let* BG *be a diameter,* *and* AD *a tangent…* *Let a straight line* JM, *parallel to* BG, *be described through* J.
1B. THE THING SOUGHT	*That* EK = EL.
2. ANALYSIS	MP = PJ. … A, N, E, D *are on a circle.* *The angles* DAE *and* END *are both right angles.*
3. SYNTHESIS	*Because the angles* DAE *and* END *are both right angles,* A, N, E, D *are on a circle.* … *Hence* MP = PJ. … *hence* KE = EL.

Q.E.D.: The structure of a geometrical proof is identical to the analysis–synthesis pattern described by Pappus. Hence, it is the written presentation of such a proof that his method describes.

(Parenthetically, the many variations on design methods here find their counterpart in geometry. The schema used by Hintikka & Remes divides each of the three steps into two parts, of which I have included only 1A and 1B here; The traditional Greek schema contained six parts: *enunciation*, *setting out*, *specification*, *construction*, *proof*, and *conclusion* (*ibid.*, p. 6). Like the various design methods, Pappus' schema and these two are variations on one and the same theme.)

How come this mistake?

Put differently, the fundamental oversight in Pappus' method is that the structure of the *product* and the structure of the *process* behind it are held to be the same. It is evident that actual problem solving does not, and cannot, follow Pappus' schema.

It deserves to be said, though, that Pappus does make certain important observations. Still, there are some fundamental oversights. These have caused serious problems, in particular because they have been carried over into other domains, in those later models that draw upon Pappus original account; it would however be wrong to hold him responsible for this.

With this clear discrepancy between what Pappus writes and what really happens, one is left to wonder how such an oversight could be made: How could the structures of process and proof be conflated? Or was it ever realized that one was mistaken for the other? Especially since the difference is so very clear, and that many others have made the same mistake later. For example, Hintikka & Remes (*ibid.*) do not detect the mistake, but even write the following in the first sentence in the opening chapter:

> Analysis is *a method Greek geometers used* in looking for proofs of theorems (theoretical analysis) and for constructions to solve problems (problematical analysis). (p. 1, my italics)

This is in a book where one chapter aims to examine whether Pappus really followed the method in producing his own proofs. And as seen earlier, Polya (1945) adopted Pappus' scheme in its entirety as a central piece in his own recommendations, in a book which is expressly meant to provide concrete help and practical advice on problem solving.

2.3 The modern mistake

But the mistake, where the structures of product and process are conflated, is not an isolated instance. The same mistake has been repeated in modern times, although in a new, updated form. This shows that it cannot be written off to the primitive state of Greek science, or as a singular incident.

As stated in section 1.2, proofs do not look the same way today, due to advances in modern formal logic. The steps are no longer presented both backward and forward; now there is just one sequence going in only one direction, forward from axioms to the proven theorem (also cf. figure 1.8).

The marks of this pattern in models of rationality are found mainly in disciplines such as software engineering, engineering design, and artificial intelligence (cf. Adelson & Soloway 1985, Dasgupta 1989, Jeffries *et al.* 1981, Parnas & Clements 1986, Swartout & Balzer 1982). This is probably due to the direct and indirect links between these disciplines and formal logic. For example, a computer scientist is much more likely to have been trained in formal logic than is an architect.

The most concise evidence of the modern pattern is the quotation from Parnas & Clements given earlier:

> Ideally, we would like to *derive our programs from a statement of requirements* in the same sense that *theorems are derived from axioms in a published proof.* All of the methodologies that can be considered "top down" are the result of our desire to have a rational systematic way of designing software. (1986, my italics)

Elsewhere they also write: "Each step taken can be shown to be the best way to get to a well defined goal." This clearly refers to formal logic, where each step in a deductive chain has been formally proven to be correct. The authors also endorse a design method that follows this same pattern.

The main features of the modern version of rationality are firstly, that the distinction between analysis and synthesis has disappeared, as two phases going in opposite direction; secondly, that the process itself also starts with the axioms, and consists in deriving the theorem from them. These changes mirror the changes in the format of modern proofs.

As a result of adapting this "modernized" pattern, the design process is taken to be a progression where the design is derived from the requirements, as though the theorem were derived from the axioms, and not the other way around. Also, the requirements specification is here held equal to the axioms (cf. the Parnas & Clements quote), when it actually corresponds to the proven theorem, since the requirements specify what is "sought". This quite serious mix-up is probably due to the fact that in design methods you are to begin with the requirements, and that the modern proof schema begins with the axioms—hence, requirements and axioms are mistakenly equated with each other.

The problem with this pattern is of course that it is an even greater distortion of the genuine process. In spite of all the weaknesses of the older schema, Pappus did make the important observation that you begin by having something you want to solve or prove, and work backward from there, rather than forward; it is also significant that you usually don't know initially just what axioms or other proofs you will make use of. (The two points are also related.) Even so, from the presentation format of modern proofs, the impression you get is that this is the procedure behind them. (There are modern approaches like proof trees and natural deduction, but which have had a rather restricted impact.)

Still, even though the case here is so open and shut, it is not different *in essence* from the distortions of Pappus' original pattern. It is the same conflation of product and process—only the effects now being more exaggerated. In both cases, the form of the presentation is based on the knowledge of how to present the proof convincingly; how to make it look rational, regardless of how it was discovered.

Today, the presentation format has been even further developed, now being even more dissimilar to the process behind it. Therefore, it is now even more obvious that this is not how it was really done. (But evidently not obvious enough.) Hence, taking this as an account of the process results in a distortion of the same kind, though to a higher degree.

Normative model = descriptive expert model

The misrepresentations in these methods are of two kinds: On the one hand, they do not work as prescriptions—people don't use them because they don't work for their advertised purpose; those who actually tried them failed to reach the stated results. On the other hand, they are also inadequate as descriptions—if you study how practitioners really work, you will find what they really do to be something quite different. For all the methods in this chapter, the failures are of both kinds.

At first thought the descriptive and normative dimensions would seem not to be related, but in fact they are, for a reason which seems to apply generally: In Pappus' case, even though his method is a "how to", he states that it is the method of three authorities (Euclid, Apollonius of Perga, and Aristaeus the Elder). Hence, his method is an account of how the experts work, to the effect that you should do the same; this is also a rather forceful argument.

Therefore, the method is a description of how skilled practitioners *do* work, and its principal function is as a prescription of how others *ought to* do their work. This is also how it is possible for the method to fail in both respects: Because it is a misrepresentation of what expert practitioners really do, the detailed procedure also fails to serve as an aid for others. Hence, the normative and the descriptive are here one and the same.

2.4 The problem is in the intramental model of rationality

The underlying pattern of rational action that is shown in the Rosetta stone (figure 1.15) also substantiates the claim that the problems with the mentioned cognitive models do not originate in the computer metaphor of mind, but in this intramental model of rationality. While the standard account in cognitive science used to be that man has been made in the image of the computer, it has recently become increasingly clear (prominently by Hutchins 1995) that the converse is the true: The computer was made in the image of the human (if not the human *mind*).

It is thus the computer model of mind that has inherited the properties of the intramental schema, and so it is a mistake to attribute the problems of the classical cognitive theories to the computer model of mind; these have just *exposed* the problems they inherited from the intramental model of cognition, going back to Pappus, Euclid, and Aristotle. To avoid these problems, it is not enough to abandon information processing theory, but to completely avoid the intramental view of cognition, computerized or not.

There is also good reason to believe that this underlying model is fundamentally wrong. Firstly, it is obviously odd that it is modeled after the structure of ancient Greek geometry proofs—the bare thought is highly improbable, but the evidence is quite firm.

But an even better reason is that the brief look at how these proofs themselves are produced shows that it is seriously wrong even about them—as was shown above, being based on a quite fundamental, open *mistake*, a confusion of product and process. Had the original model of mathematical problem solving been good, then it may quite possibly have been a good start for a theory of action and cognition in general.

For the remainder of this book, the most important connection between design methodology and cognitive science is this failure, since it is located in the model of rational action; on a *cognitive* level, that is. Thereby this failure becomes of general interest; design becomes a domain where the underlying model of rationality has been put to use, under highly authentic circumstances, and failed.

The remaining chapters will follow this path, looking at design processes in order to locate and explain this cognitive failure, but more importantly, to also present a better model of design, and of sensible reasoning and action—the one that Euclid by a certain baneful craftiness kept to himself.