

My Books

I enjoy writing and I enjoy drawing.

So I usually combine the two.

I have had two books published.

Most authors publish one, and stop at one for various reasons.

Or they go on to publish many.

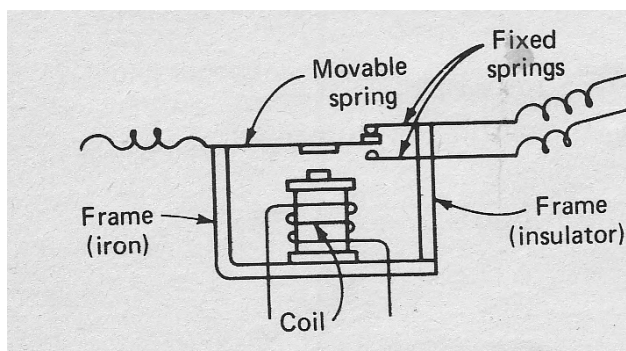
But very few stop at two.

"Digital Systems Fundamentals"

was my first book. It was an introduction to Engineering systems from a discrete or digital point of view.

It had a unity which came from simplicity.

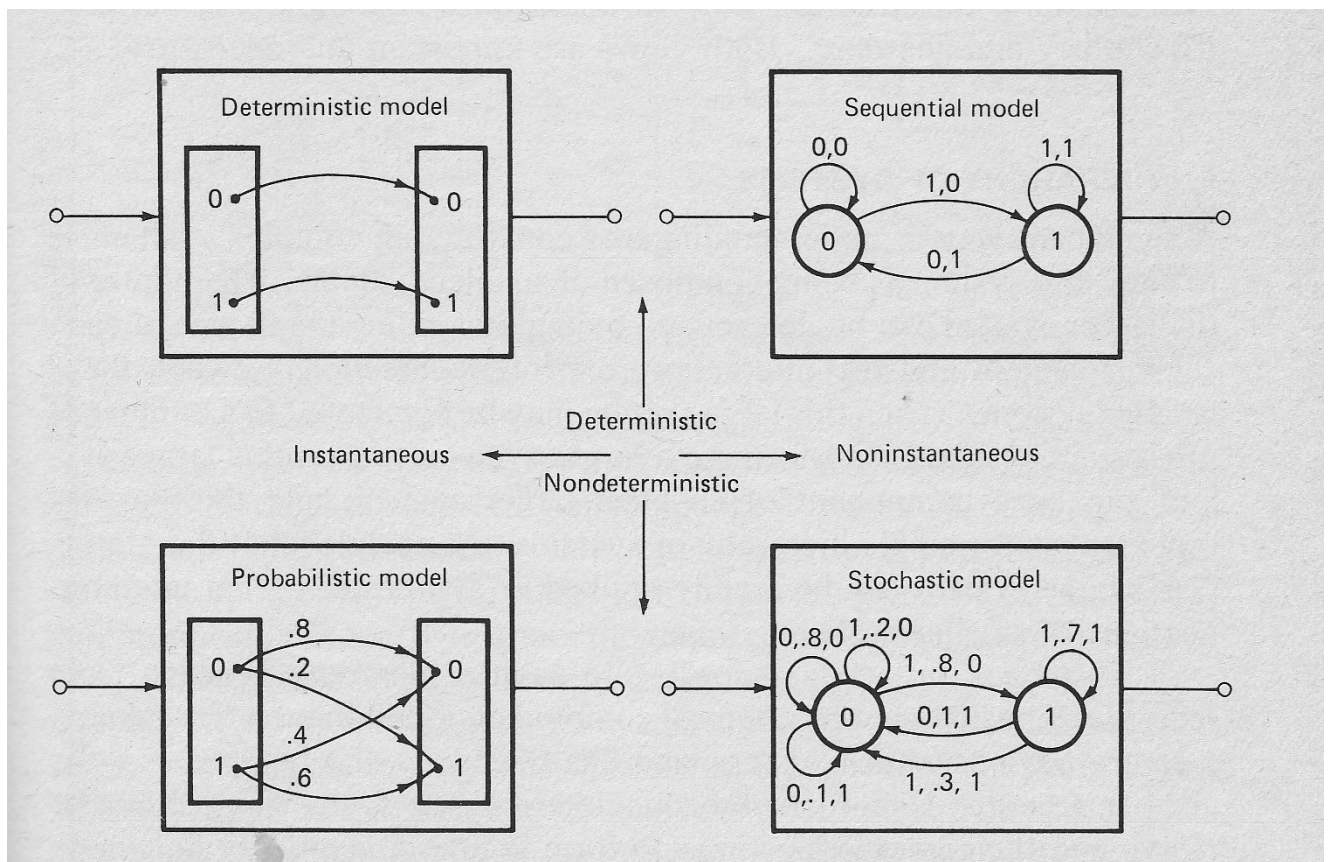
It began with an electromagnetic relay which is a coil of wire that when energized could attract a spring and make contact.



This leads to four models of system behavior;

when voltage is applied to the coil:

1. the contact always closes doing so instantly,
2. the contact usually closes, doing so instantly,
3. the contact always closes, but after a delay,
4. the contact usually closes, but after some delay.



Each of these models has a corresponding diagram and mathematics:

(Boolean algebra, probability, automata theory, markoff chains).

These models were applied to other systems besides relays

including mechanical, fluidic and electronic systems.

System processes were developed for each model (via Bloom's taxonomy) including: modeling, identification, representation, analysis, synthesis, implementation, optimization and evaluation.

Because of the simpler digital nature of these systems, it was possible to cover at least 2 of these 4 models in a single semester, compared to 2 or 4 semesters for the ordinary continuous systems (along with calculus, differential equations, etc).

This book was published by McGraw Hill, with an international paperback edition. I also created a set of 30 video-taped lectures, which in the 1970s was more of a challenge than in these times. The book is not obsolete because it emphasizes logical fundamentals, rather than the latest devices.

I found the publishing process at that time to be horrendous with its galley proofs, constant corrections and rewrites, I did not want to do that again. So when I felt inspired to write another book, I first wrote a text processor, and also a diagram processor in order to present a camera-ready manuscript to a publisher.

"Programming Principles" was my second book.

It too concentrated on principles but was geared toward software.

The main chapters were devoted to programming concepts in general, but there was an appendix chapter devoted to the Pascal language.

It too is not obsolete, as it deals with principles common to many languages, and was taught with Fortran, Modula, C, C++ and Java.

The structure of this book was unique in that it consisted of page pairs, where each left side had diagrams and the right had text only.

It appealed to both the left and right brain equally.

It was ultimately published by Allyn & Bacon and later by WC Brown.

I also wrote, but did not publish, a book on programming principles with each general language-free chapter alternating with a sub-chapter on the 'modular' language Modula-2.

Unfortunately, it takes me about five years to write a book, test it in my classes, and continue revising and refining.

During this time other authors have published some updated versions. And then the field moves on to another language.

I also extended this language-free approach to programming while working for Metrowerks in Montreal (to publish with Harper-Collins). They, and many faculty, had little interest in a language-independent approach and so it never did get published (in a paper form).

Most faculty prefer to introduce beginners to huge, industrial languages (like C++, or Java); I feel that it's like teaching a beginner to drive using a race car or a military tank. It can be done, but it has side effects (student confusion, poor error messages, distractions from principles).

This book (all 525 pages) with its language-independent approach can be found on the internet at the Russian university MIPT site:

<http://cs.mipt.ru/docs/comp/eng/develop/common/principles/main.pdf>

My name has been removed from it, and the only proof of my involvement can be seen on page 179, where I used my name and city in an example of a record.

The book still used many diagrams to illustrate concepts, but I also developed a general pseudo-code as a language. The new Python language has a very similar structure to my pseudo-code (just insert a colon after each structure such as: if, else, while, etc).

I wrote detailed course notes on other subjects but never published them (if only the internet was available then). Subjects I wrote about include: linear systems theory, micro computers, information theory, multimedia, and many-valued logic.

I also created a learners language called Jr, with a colleague from CalTech. Jr is a 'junior' version of Java, Eiffel, and Oberon and it ran on an online Environment, JJ, but that is another story (see "My Languages").