# *Pendies for Lops*

***by Jan Karman***

## Abstract

Participants of a pensiunfund can simulate the effects of postponing or accelerating their retirement date, exchange of types of their pension, make salary increments and change the parttime percentages. It was initially designed for PR-purposes, but now it's also used in our actuarial department for making quotes for our participants.
The presentation tries to show what a single APL developer can do in 6-8 weeks.
The software can be used as a template to similar implementations for (large) modern pensionschemes.
Keywords are: 100% APL;layman's tool;fast and safe development;FUN

------------
The name *Pendies for Lops* was invented by the researcher of Moret Ernst & Young, Holland who performed a comparative research on products of its kind in The Netherlands and in Quebec, Canada. In English you could say: Pension Diskette for Laymen on Pensions.

## History

One of the tasks of a pensionfund management is to detect trends in society and in the market in their field, translate them into comprehensible pieces and in well formulated proposals to the Board and the Participants Meeting, in order to get them accepted as modifications to the existing regulations. Pensionfunds used to tend rusty institutions in terms of dictating the schemes, until 1995. Then a trend to flexibility emerged and the large companies like Philips, Shell, as well as the Government designed modifications to their schemes in which e.g. the date of retirement became flexible.
Our management started these "campaigns"  in December 1995 with a proposal to the Board.
In February 1996 the rules in general were accepted and a time track had to be settled. There are quite a few stations to be passed before such proposals turn into "Regulations". The most important are the (larger) individual employers (EMPs) involved (in our case a few tens outof a few hundred), a Committee on Labour conditions Agreements (CLA), the Unions, the Companies Councils and - at the terminal - the Participants Meeting (PM) and the Founders.
This time the changes were so drastic that we decided to have a diskette for the PC which would clearly show the impacts of the changes in any individual case. I promised to have a diskette in four weeks, although I didn't know how, yet. I called a meeting of the most experienced programmers in our IT-Dept (for which I have been responsible last 13 years) and asked volunteers for the job. Nobody stood up. So, I had to do it myself. (You cannot manage to force people for such a job, that would not work).
The time schedule was very tight, the PM is a yearly meeting at a more or less fixed date in June and we would allow people to have a two weeks to play with the diskette. Early during the track it was desired that we should also have a Macintosh-version. The software house who was supposed to finish this would need 6 weeks. So, end of April was sort of deadline.
On the 22nd of March I had a first version ready for presentation to the CLA (note: based on preliminary specifications!). The CLA wasn't content with the new rules and the outcome was that a big part of the specification had to be changed.
On April 15 the new rules were settled by the Board and on April the 22nd 200 diskettes could be shipped for testing. The last week of April most of the time left was spend on presentations. Only some polishing and the installation procedure were left to be done and on May 22 the individual statements were mailed, together with 8000 diskettes.
The regulations were accepted on June 19 by the PM unanimously.

## The nature of a layman's tool

If you're building an application for users in your company you can afford some bugs. Even, be it to a lower degree, a product for developers (like Dyalog APL) can afford some - minor - bugs.

If  you’re building an application for laymen in order to avoid hundreds of phonecalls on questions about the impact of changes of rules on people’s money it better be sterile. Needless to say that also the installation should be smooth. And last but not least there should be an online “Help”.

**Why APL?**
The gadfly-question. Well, I don’t probably have a satisfying answer for most of you.
Speaking for myself the answer is unusually uncomplicated: it’s the only language I can make decent software in - I’m not a programmer. And since I was the only person left to fix the job it became APL, one hundred percent APL, that is.

David (Siegel) gave me some hints for sub-headers, which I found very useful.
I’ll follow them globally here after.

**How did APL ease our develoment process?**
I have no comparison at all. I only know that I can usually finish jobs in APL quickly and neatly. APL in our department is exclusively my (the dept.manager’s) domain. Production software used to be done in (generated) COBOL and nowadays in a Uniface / Solid environment. I only discuss and judge the arguments for purchasing a development tool and leave the rest to the programmers.

The other thing is that I can read actuarial stuff, as used in practice that is, and I can read and write APL. So, I could program at the speed of thought - really not a cliché here.
Further I think I can compose rather complex applications pretty well. All together APL worked out well here and keeps doing so succesfully in hectic situations where (actuarial) computation is involved.

**How did we design our application in a GUI enviornment?**
To be honest, this was one of the keys to our success. This is not a place to sell a particular product, I assume. But I had already some experience with the Causeway GUI-tool, which comes with the Dyalog APL/W interpreter.
Causeway has a designer by which you can “paint” your screens just by mouse clicking, assign variable names to objects, telling them how and when to refresh, depending on which change of what and decorate them.
The designer has an “Event Action Table” (EAT) department, in which you can be a fulltime dedicated APL-er and a “Data  to Watch” (DtW) department. These two departments are sufficient to design the “whats, whens and hows” of the entire application. The EAT is “broadcasting” the new values according to the prescriptions in your code and the DtW is “listening” to this radio (or rather “telegraph” - it’s not wireless) changing subsequently the display as applicable.

**What hurdles did we encounter and how did we overcome them**
The *first* problem was that in such projects there are no written specifications. For the Actuarial Dept. everything is new also. You even don’t globally know what’s involved.
So, you start from iota zero. In the beginning you only have your general experience with the métier and your creative fantasy.
What could possibly be in it? How should it look like? And you just start staring at your screen. There is an *old* situation and there is *new* situation. How to come from here to there? Which variables are involved? Continuously be aware of how stirdy and robust it has to be.
When it came to the code we sat with two people at my desk, our actuary and myself.
He was dictating the hand-written formulae and I was translating his mutterings in APL code.
Who was talking about methods? The things-getting-be-done method. Spinach!
Then, from time to time we had the noisy entering of our Manager: “Could you also add something for the women who already have the option for leaving at 60?”
*Second* was the GUI-development tool: Causeway. This is not contradicting the earlier statement.

The Causeway product was very young and my experience with it was little. These two factors resulted in detecting - sometimes fancied - bugs. I must say that we got excellent support from Causeway.
*Next* there was the testing.
As soon as the diskette got some substance everybody in the office was hold to test the program and was supposed to make reports of it - in the weekend. This is a frustrating part. You will learn that only one or maybe two outof say 50 are capable to do valuable work. That is, capable of detecting errors by using the stuff systematically on the one hand, and writing down the experiences on the other.
Most of them say: “Nice program” or call you every minute with: “This is not working...”
In the end however when there are only two really good testers reports you may assume that 90% of the bugs were recognized. This testing involves validation of input, order of the cursor through the screen, consistency in location of the controlls, logic of the menu structure and so on.
We also asked Causeway - then Adrian & Duncan - to do a “Workspace Health Check”. A project like this is a NLG 50.000 (2 to 1 $US) operation and it really pays off to have an outsiders’ judgement. It certainly did in our case.

The sting was in the *tail*.
We had targeted the diskette for use on Microsoft Windows, then 3.1x. Some users had already Windows 95. And either those users had sometimes problems with the OS or with the software, it was sometimes hard to detect. Quite a few had problems with their too light PC in the first place. We got angry letters from those. Also people called disturbed, complaining that this was “just  pollution of the environment”. One person wrote how I could “have the guts to display Microsoft logo on the ‘About’ plaquette” (it appeared that his son was studying Computer Science).

In general, however, the operation was a succes, and the outcome of our initial inquiry was that 80% of the responders were highly satisfied and 10% was explicitly dissatisfied.  The remaining were in between.

**How did we distribute our  application?**
For the installation software part we used WinBatch. In the available version real Windows unpacking of the files was not possible. We had to make use of a Call to PKUNZIP and this resulted in a less neat performance of the unpacking. We usually don’t make software for distribution, so we had no experience with that aspect.
This changed when we moved to InstallShield. The installation looked professional from that time, but the overhead was far more than with WinBatch: it now needed a second diskette. We thought it worthwhile nevertheless and we released our Version 2.0. The only difference with Version 1.0 was the installation tool.
As soon as the diskette was cleared for release a “master” disk was sent to a professional disk copier, who als took care of the shipping after testing a randomly taken sample. The shutters carried our logo and it was wrapped in a neat soft plastic cover with some text.
On the label it said: Installation: **Run a:\setup**.

**Continuing story**
We were incented by the succes and now we have our Version 3.0 which is quite different from Version 1.0. The first version was mainly aimed for illustrating individual effects caused by the changes in our pension scheme and for promoting the fact that we were on the right track in the pension market, and so doing, try to convince all the institutions involved for the good of it.
Version 3.0 represents the rules of our new pension scheme from an individual point of view. It can be used by the individual participant when at home, but also by our actuarial staff as an interactive tool for enlightening the complex matter of pensions to our participants. It also holds a simple but effective database, controlled by way of a *combo box*, which makes it also useful for the personnel departments of our enrolled employers (about 200 now).

Hilversum, The Netherlands,  October 17, 1997