The Whittle Story
A Live Book

Working with Jeff Whittle

By Chris Wharton, November 2019

This is a brief overview of working with Jeff while working on Four-D, Four-X, Opti-cut and the Whittle graphical interface (GUI). Many papers have been written on the workings of the products and some are referenced at the end of this article.

I joined Whittle Programming in February 1993 when Jeff was working on Four-D. His initial product, Three-D, was already making steady sales and this next product was tackling the big issue of the time cost of money and its effect on the size and shape of the final pit. The manner in which you mine a resource has a huge impact on the net present value (NPV) of the mine and indeed Jeff has continued to work on that theme for over 25 years.

The rise of Whittle Programming occurred at an interesting time in the evolution of computing hardware. Mining companies were using workstations built by DEC, HP and SUN, IBM had released a personal computer and Bill Gates had released a computer running MS-DOS (version 3.1 when I started). Jeff was using PCs to write and support his products which were written using FORTRAN IV software. FORTRAN (FORmula TRANslation) was very popular for mathematical computations. The optimisation number crunching is still done using FORTRAN. The FORTRAN compiler changed over time with operating systems, a migration to FORTRAN 77 and other technical issues. We used a range of compilers including Compaq, DEC, Lahey, Salford, UNIX, VAX and WATCOM. The source code contained compiler specific instructions and there was a pre-processor to set the routines up for a specific compiler.

These days we take computer memory and storage for granted. We install programs on our smart phones and expect graphics and instant user interaction. Back in the mid-1980s the initial release of Three-D was a text-based interface. The screen typed out a prompt and you entered a response. Memory on a PC was limited to 640 Kilobytes. Storage was 512 kilobyte floppies and small hard disk drives. These limits affected software design and meant that complex software often had to be organised so that intermediate calculations could be stored and recovered. Jeff had already created virtual memory routines that would use hard disk storage if there was insufficient actual memory to store the data. Retrieving data from storage was far slower than retrieval from computer memory. Virtual storage may be needed when loading mine models and was particularly noticeable when doing optimisations. In the early days block models were small and optimisations could run for hours. The block models were based on information from the geological databases and provided spatial grade tonnage information. The format for the block model interface was defined during the earlier Three-D work and it would be enhanced over time as more options became available.

One way of speeding the work up was to regroup the block model by combining blocks in any or all directions. You could also define a default waste and only load blocks with ore. These tricks required less memory and could keep the program running in memory rather than using the virtual memory. Jeff had already built in a restart option on the optimiser and you could also save the results from the optimiser at specified time intervals. This was useful if there was a computer crash or the mine site had flaky power supply.

Jeff was very meticulous in his methods. His software was documented well, and he had details on all the subroutines that were used. One of the limitations of early versions of FORTRAN was that you could only have a maximum of 8 characters in a variable name. Jeff had structure in his naming conventions. He used four-character names for programs. The first two characters were the product and the next two the module. For example, FDOP, FDAN for the Four-D optimiser and analysis programs. The subroutines were six characters in the form of product, verb and object. For example, FDCHPA CHecked the PArameter file for Four-D. This made it a lot easier when writing the code to remember the subroutine names.

The development was on PCs, however, the software had to run on the machines that the mining software companies were using. Jeff was using companies like Gemcom, Maptek and Surpac, as agents, and they were on selling his product as no one else had developed their own Three-D (Lerchs Grossman optimiser). This meant that updates and bug fixes had to be able to run on the workstations. There is an old joke about standards “the great thing about them is that there are so many to choose from”. Simple things like time and date varied from machine to machine across the workstations. And at times there were differences in the FORTRAN compilers on the different workstations. The task of updating required access to the workstations. Jeff had developed some test routines to try and test the new releases. Over time, as the software became more complex, the suite of test routines grew and was probably near 500 when the company was sold to Gemcom. The test routines were invaluable and greatly added to the reliability of the software. Whittle had very few releases that needed immediate corrections.

Jeff was also very scrupulous with bugs. If there was a bug that impacted the validity of the results, then that would demand an immediate fix. If there was a work-around, then all existing clients on technical support were advised. If not, then the software was updated as quickly as possible depending on the platform and a new release was issued.

Jobs and Wozniak started in a shed. Ruth and Jeff started in their home. Upstairs Ruth was marketing, David was business manager, Philippa Manz was secretary and Jeff and I were beavering away on upgrades to Four-D. Over time the office expanded to Balwyn sometime after Geoff Hall joined as another programmer and later to Box Hill as more staff was taken on.

Four-X was an extension to Four-D that allowed multiple elements to be carried in the model file. Prior to that additional elements had been treated as equivalent metal. The new version allowed each material (element) to have its own price, recovery and selling costs and reporting. A single element licence still used the Four-X software but had a maximum of one element allowed.

Four-X was constantly improved. Clients that had software support got the new version for free. Over time more features were added:

- Four-X Expressions: These were expanded to allow better definitions of process recovery, Excel style and/or/if structures, access to all block model values (index, location) etc. Fixed cut-offs are no longer strictly valid when expressions are used that affect the product price, recovery or cost of processing so cashflow optimisation was introduced to determine if a parcel was processed or sent to waste (or stockpile).
- Milawa Algorithm: Improved scheduling options Maximize NPV or balance material handling to allow a more even mining rate without destroying the NPV.
- Minimum Mining Width: Allowed pushbacks to be modified to provide a user defined minimum mining width and make the mining schedule more realistic. The user had full control of the tweaking features and the software ensured that slope integrity was maintained as changes from lower levels were applied to higher benches.
- Blending: Material taken directly from the pit may be combined with material from one or more optional stockpiles, to create one or more desired blends, each of which is suitable as an input to a process. The in-pit material is represented by a number of bench/phase panels, each of which has defined tonnage and grade characteristics. These panels have a defined availability depending on their position in the mine and the defined mining sequence. Material in stockpiles is likewise represented by tonnage and grades. The combination of panels and stockpiles forms the basis for a Linear Programming (LP) Optimisation. The blend optimiser seeks to maximize cash flow from the processing of the available panels while taking user constraints into account.
- Multi-Mine processing: Allowed mines to be scheduled independently and each mine could have different phases.
- Directional mining: This allowed mining to proceed in a specified direction and was controlled by expressions.

The Four-X product actually comprises 6 modules:

- FXED - parameter file editor. Essentially defunct with the advent of the GUI
- FXST - creates the structure arc file to support the mining slope requirements
- FXOP - Carries out the optimisation in either cut-off or cashflow mode
- FXAN - Performs all the scheduling analyses
- FXUT - Utility program. Reads model and results files and passes information back to the GUI
- FXMI - Mining width processor. Provides specified minimum mining width at the pit floor and alters phases on higher benches by applying structure arcs and user constraints

Opti-Cut development started after the changes to Four-D and before the development of Four-X. As explained earlier, Four-D took time costs into account and you could change prices, costs, recoveries and equipment capacities over time. The decision as to whether a parcel of material was ore or waste was based on the marginal cut-off. Opti-Cut was designed to look at changing the cut-off with time to maximise the NPV from the mine. Any material that was below the revised cut-off could be sent to stockpiles or to waste if it was below stockpile grade limits. The original inspiration for this work was the book by Ken Lane entitled “The Economic Definition of Ore”.

When mineral prices increase, the economic (break-even) cut-off can be dropped but this merely puts more low-grade material into the mill. Lane explained that there was an opportunity cost that had to be considered in addition to the costs associated with the cut-off. This could lead to the elevation of cut-offs to make use of the improved prices. Conversely when prices drop you should consider lowering the cut-off to preserve the resource for better times. In the end we used none of the optimisation techniques of Ken Lane and instead used a ‘delay cost’ and a ‘change cost’ developed in-house.

Taking a concept and turning it into a commercial product takes a lot of time and R & D effort. The work was developed as a standalone program. A lot of development code didn’t make it into the final version. Opti-Cut probably had two years of development before we released initial beta test versions. Opti-Cut also uses a set of standalone programs to perform the various functions of setting up the analysis, performing the optimisation and reporting the results.

Jeff was pivotal in providing the mechanism to search for the improved cut-offs. There was also a lot of work in making the process efficient and speedy and ultimately in allowing the results from Opti-Cut to be fed back into Four-D/X to produce traditional schedules. Opti-Cut also drove the need to work with stockpiles and have all the associated limits, cut-offs, reporting and spreadsheet codes.

The porting of code from Microsoft based FORTRAN to the workstations used to take a few days depending on the availability of machines and occasionally it involved a trip to Perth where we had access to all three machines. Jeff used to do the work but then passed the baton on to me.

Windows based PCs were getting a lot faster and the cost was significantly smaller than a workstation. Within a very short time-frame mining companies were starting to use Windows

based computers instead of the workstations and our primary delivery was to the PC market. However, we still had to solve the issue of introducing a graphical user interface to make our product more acceptable for the times. We investigated several products which purported to allow code to be written once and ported to other machines, but the initial cost was very high and it only solved part of the problem.

Java was initially released in 1996 and version 1.1 in Feb 1997. It promised the concept of portability where you could write once, and it would run on any machine with a Java compiler. We were initially sceptical and wary of basing our code on a new technology. However, new releases were being issued frequently and all the major vendors were starting to adopt it. Darren McRostie came on board and led the initial development work. We were definitely testing before release 1.2 came out in December 1998 because we upgraded to the new Swing graphical API which was introduced with that release.

Consideration was given to porting the Whittle FORTRAN code over to C++ but after testing the interoperability of running FORTRAN programs from Java we adopted the position that it was safer staying with FORTRAN based on our huge test set and we were better off putting our efforts into the GUI. We worked out ways of getting the results back from the FORTRAN code to show it in the GUI. The process was seamless, and users are not aware of the work done in the background.

The new GUI also introduced the use of a Three-D visualiser for block models, results files and mining sequences. This allowed the user to visualise based on pit shell phases, benches, XY, XZ and YZ views as well as fully rotational pit shells with or without surface topography.

Technical support was mainly carried out by Geoff Hall and myself. Over time we implemented a bugs database and this was used to provide input to the technical releases. This also gave us a valuable resource of what issues were affecting users.

Whittle provided training locally and abroad where required. The main trainer was Norm Hanson. A geologist by initial training, he was a font of mining intelligence and a really good tester of the Whittle software. When Norm found a bug it was normally a good one. He was also responsible for the Marvin gold copper test data set that was used in many of our tests and has now been used by many other developers testing their results against Whittle's.

The company was eventually sold to Gemcom and Jeff continued to work on scheduling and improving mining optimisation with Whittle Consulting.

Papers published on the products

Opti-Cut

- Optimising Cut-Off Grades - Jeff Whittle and Christopher Wharton
- Proceedings of the 1995 Optimizing with Whittle Conference Perth 13-14 March 1995

Four-X Expressions

- Using Expressions in Four-X - Chris Wharton
- Third Biennial Conference STRATEGIC MINE PLANNING Perth 23-24 March 1999

Milawa Algorithm:

- Add Value To Your Mine Through Improved Long Term Scheduling – Chris Wharton
- Whittle North American Strategic Mine Planning Conference, Colorado, August 2000

Blending

- The Use of Extractive Blending Optimisation for Improved Profitability - Chris Wharton
- Proceedings, Orebody Modelling and Strategic Mine Planning, AusIMM Perth, 22-24 November 2004

Multi-Mine processing

- Development and application of Whittle Multi-mine at Geita Gold Mine, Tanzania – T Joukoff, D Purdey and C Wharton
- Proceedings, Orebody Modelling and Strategic Mine Planning, AusIMM Perth, 22-24 November 2004

To read Chris' bio please scroll to the next page..

Chris Wharton

Chris worked for Whittle Programming from February 1993 until December 2001. Over that time, he worked as a Senior Systems Analyst and Technical Consultant on mine schedule optimisations, the implementation of Opti-Cut and the formulation and implementation of 'Cash Flow Grades'. Chris has worked on all the Whittle software from Four-X, Opti-Cut, blend modules and the 3D visualizer.

After the sale of Whittle software to Gemcom Software International, Chris worked for Gemcom from January 2002 until March 2006 as the Whittle Development Manager. In this role, Chris was responsible for the multi-mine module, cash flow grades via user-defined elements and user-defined block model evaluations.