

Twenty-eight years later, Steve Milligan reflects

I always thought the “great breakthrough” was the notion that multiple analysts could have random access and share an intrinsically sequential and non-shareable resource (multiple tape drives). The sequential, non-shareable nature of tape drives is why Hughes was stuck on batch analysis. They couldn’t imagine doing anything except read tapes front to back with a dedicated program. The notion of the DataCache (and the Tape Slaves feeding it) allowed both shared and random access by multiple analysts simultaneously. This and the Data Dictionary made everything else possible. In particular, decoding the launch sequence peculiarities was impossible without random access. Of course, eventually disk storage caught up with tape capacity and one could have just copied everything to disk, but back at the beginning there was more data than anyone had ever considered for random access.

tape-resident data, but independent Tape Slaves are still critical for dealing with the disparate data streams and time anomalies.

Milligan also devised a means of representing the plethora of recording modes in a *Data Dictionary* that could be modified³ as new modes or errors were encountered, without changing the underlying program. This proved to be another critical innovation, keeping the unpacking and display of data independent of one another and enabling rapid adaptation of DataProbe to new data sources and recording systems.

A few years later, Ben Dubrovsky expanded the data-dictionary approach to create the *Flexible File Server*, which enabled user-configurable access to an entire data set on disk or tape: physical and logical records; record lengths, IDs, and time tags, as well as the individual variables stored within. This capability was critical to commercial success because it enabled a support engineer to quickly connect DataProbe to a customer’s data during a single sales call. It was later modified to handle real-time data, with the simple artifice of polling a data source and adding data to a growing file.

The beauty of this architecture—autonomous tape control, multiple levels of caching, and a program-independent table of record structures—was that analysts had only to think about specific variables of interest; all details of the data extraction process were conveniently invisible.

4. User Interface and Data Display

At the same time, Fortmann and new hire Jim Arnold were implementing the more visible components of the software, basing the user interface on a command-line-interpretation library called COMAND. With roots in the TENEX operating system (Chapter 22, section 2.) and the BBN Speech Group (see sidebar), COMAND was developed in earlier sonar signal processing and tracking projects and extended/refined for DataProbe.

The user controlled DataProbe (and the earlier sonar programs) by means of a novel command-line interface with automated command recognition: typing “?” would display all available choices, “escape” would fill in a command or subcommand, and “noise words” indicated what input was expected next. Jeff Berliner had designed a clever COMAND-based utility called PARCHG (for PARAmeter CHAnGe)

³By means of a separate, interactive, Data Dictionary editor program.