CELEBRATING 30 YEARS OF NATURAL PROGRAMMING LANGUAGE
NOW MORE VALUABLE THAN EVER

By Karlheinz Kronauer, Director Product Management, Software AG

When Dr. Peter Page and Margit Neumann developed the fourth-generation programming language (4GL) Natural in the mid 1970s, nobody would have dared to predict that they were laying the foundation for a success story that continues even now. Meanwhile, Natural has become a modern development environment and is in use at numerous companies around the world. The programming language is now a state-of-the-art interactive development environment that quickens the pulse of software developers.

Here’s how it all began: Natural’s inventors were given the task of calculating dog license fees for the city of Vienna, Austria using the Adabas database system and the COBOL programming language. Assemblers were still frequently used for programming in those days, and the third-generation languages COBOL (COmmon Business Oriented Language), FORTRAN (FORmula TRANslator), and PL/1 (Programming Language One) were considered state of the art. Experience gained on this project thoroughly confirmed Edsger Wybe Dijkstra’s criticism in his 1975 article, “How do we tell truths that might hurt?” The Dutch computer scientist and Turing Award winner bitingly wrote that “the use of COBOL cripples the mind; its teaching should, therefore, be regarded as a criminal offense.” And that was the inspiration for developing Natural.

The first version of Natural was released in late 1979, with the city of Vienna as the first customer. “Within two months, we completed the first application and started using it,” recalled Peter Leibl from Vienna’s data center.

What is so special about Natural? As the name makes clear, ease of use is the highest priority. Programmers can focus completely on solving business problems without having to deal with complicated syntactical constructs and platform-specific quirks and tools. Natural’s second principle for success is to “write once, run everywhere.” Applications developed with Natural run on almost all hardware platforms, from netbooks and Linux to mainframes. The syntax based on natural language simplifies learnability and maintenance, especially for English speakers.

Natural was originally developed for mainframes. In the late seventies, teleprocessing monitors, remote data processing, and transaction processing were still exotic-sounding concepts at many companies. Batch processing was the norm for essential applications. Punch cards held the source code, and there were few suitable editors or tools for developing programs interactively. In particular, application programming with screen input and output for end users was still in its infancy. Syntax errors were not recognized during programming; rather, they first came to light during compiling. Even when the program code was finally complete, the linkage steps still had to follow. It was only then that testing could take place, which slowed and complicated the development process greatly. Natural eliminated this unnecessary overhead in one fell swoop, and 1979’s Version 1 included an interactive program editor. Being able to enter program code online, check syntax immediately, and instantly see the results on the screen was sensational back then and drastically increased productivity.
Natural’s advantages soon won over many companies. In the early eighties, Natural was in productive use at more than 500 companies worldwide, and subsequent development proceeded rapidly. While the 3GL languages hardly had any notable developments, the release of Natural Version 2.1 in 1987 was a quantum leap forward. Natural’s full-screen editors allowed code to be entered elegantly, and screens using character-based 3270 interfaces (a de-facto standard of IBM) for input and output could be created easily using an integrated WYSIWYG (what you see is what you get) editor. The screen editor supported the separation of processing and presentation logic, so that the programming logic did not have to be modified when the presentation was changed. And long before the introduction of object-orientated programming, Natural already had handy array editors to separate the data from the program logic, enabling reusability. The concept of structured programming, now widely used, was supported with Natural 2.1 in its structured mode, which enabled developers to maintain the rules of structured program syntax. In particular, internationally-active companies valued the opportunities presented by Natural to create multilingual applications very easily. In the meantime, add-on products extended Natural’s range of functions: Natural Security guarantees the security of the development and data-flow environments, and Repository Predict documents and manages data models and metadata. In the mid eighties, Adabas and other popular databases such as VSAM, DB2, and IMS/DB became supported.

With the expanded, large customer base, the need for communication between customers also increased. Natural user groups took root in many countries and the Natural Conference was launched in the United States. The founder of the annual conference, Jim Wisdom of Boston University, says: “The conference offers customers and prospective customers alike an excellent technical forum for exchange. Customers present their Natural solutions and discuss Natural functionalities and developments with Software AG developers.” Natural has always been geared toward customers’ needs to develop business-critical applications simply. It is always open to the customers’ requests, which are taken into account during development.

Personal computers really began to take hold in the early nineties and, consequently, the first version of Natural to run on Windows 3.1 was released in 1994. Along with the graphical user interfaces (GUIs) for the developers themselves, GUIs and event-controlled applications could also be developed with this version of Natural onwards. See Figure 2.
In 1992, the first Natural version for UNIX came onto the market, and the first Natural Linux version followed 10 years later. The Natural systems on open systems could communicate easily with Adabas via the Natural language elements or SQL syntax and with the relational database systems common on these platforms, including Oracle, Microsoft’s SQL Server, Sybase, and IBM’s DB2. With the introduction of Natural on Linux, companies began to migrate from mainframes to the Linux platform, which posed no problems, thanks to Natural’s portability. The Salzburg state government was one of the first customers to migrate successfully. “We achieved considerable cost savings with the same or better performance following the problem-free migration from Adabas and Natural to Linux. The migration was a complete success, thanks to the portability of Natural and Adabas,” says Josef Scherndl of Landesinformatik Salzburg. In the mid nineties, an additional Natural innovation was welcomed by the market: Natural’s single point of development concept. While programs are developed in Natural Studio in a Windows environment, the data-flow environment is on a mainframe, Unix/Linux server, or Windows Server. This allows Natural developers to work with all the advantages of a GUI in Windows, but the programs can be on a mainframe, where they are compiled and executed. Graphical tools support the developer: A debugger enables convenient testing of mainframe programs, and database content can be seen with the data browser.

Web-enabling features support the increasing spread of the Internet. Natural was the first language to directly process XML documents. New language elements allow the parsing and serializing of XML documents. By supporting HTTP(S) protocols, Natural programs can communicate with the World Wide Web and efficiently read and write all types of documents via links (URLs). Of course, Natural now has language elements to process Unicode data and convert code pages. Accessing SOAP-based Web services is just as simple, and, in turn, Natural programs can be made available as Web services. See Figure 3.

FIGURE 3: Developing and Testing a Natural Web Program in Eclipse
With the increasing spread of Eclipse as a development platform—primarily for Java applications—Natural was also integrated into Eclipse. Therefore, developers who are less familiar with the features of a mainframe can also benefit from the advantages of Natural as a language for solving business-critical problems.

Because of the growing popularity of Eclipse as a development platform—especially for Java applications—Natural was integrated with Eclipse. This makes it easy for developers who do not have comprehensive mainframe knowledge to develop modern, Web-based business applications.

These days, Software AG has further enhanced its support for Eclipse by simplifying the generation of Web Services and the development of AJAX-based rich internet applications. This enables existing Natural programming logic to be easily reused in browser-based Web applications. To facilitate the integration of any enterprise applications, as for example SAP, and to guarantee SOA governance, specific solutions from the webMethods suite, like Integration Server (ESB) or CentraSite, can be directly incorporated into the development environment. The new development environment is available under the product name NaturalONE. With NaturalONE developers can program Web applications and Web services which can be executed on all major platforms like mainframes, UNIX, Linux, Windows and OpenVMS.