

;login:

THE MAGAZINE OF USENIX & SAGE

August 2003 • volume 28 • number 4

inside:

CONFERENCE REPORTS

2003 European Tcl/Tk User Meeting

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

2003 European Tcl/Tk User Meeting

**NÜRNBERG, GERMANY
MAY 30 AND 31, 2003**

Summarized by Clif Flynt

THE STATE OF TCL

Andreas Kupries, ActiveState

Kupries described the status of the current Tcl release as well as ActiveState's current offerings. He reported that the use of Tcl (based on downloads) is still growing, with several major manufacturers joining the Tcl ranks.

The 8.4 series is now officially closed to new features, though still open for maintenance. (New features are being added to the 8.5 release.) The 8.4 release included support for the Virtual File System, which allows a zip or tar file to be used as a file system, and supports mounting an HTTP or FTP site, several new high-level widgets (including paned windows, labeled frames, and a spin-box), support for 64-bit integers and file systems (even on 32-bit systems), and many performance improvements. The Tcl core has been Unicode compliant and thread safe since the 8.1 release. Performance was improved with the 8.4 release.

The Tcl maintainers are continuing to improve the Tcl core, and several TIPS are scheduled for inclusion in 8.5. The new dict data structure has been added already, and more options for the expr command and more high-level widgets are planned.

ActiveState has continued to support Tcl with the introduction of a Tcl Cookbook series of code snippets, with the Tcl Dev Kit software development tools, and by continuing to extend the Tcl support in the Komodo IDE, including incorporating a new GUI builder.

NPS PORTAL MANAGER AND NPS CONTENT MANAGER

Thomas Witt, Infopark

Witt described Infopark's content management system, NPS, and why and how they decided to use Tcl as the application scripting language.

The NPS package is written in Objective C. They extend the application's functionality by embedding the Tcl interpreter because:

- Tcl is open source, using the BSD license, and so is free for commercial use.
- The same Tcl code runs on all platforms.
- Tcl is a normal glue language, with no surprises for the casual programmer.
- It's easy to learn. This is particularly important when teaching Tcl to nonprogrammers.

Witt described the basics of integrating new commands into the Tcl interpreter and demonstrated how the new commands are accessed from a Tcl script.

One extremely useful feature of Tcl is the trace command, which allows NPS to support dynamic callbacks based on changes to a document. This is equivalent to using a database trigger.

CREATIVE USE OF THE TEXT WIDGET

Arjen Markus

Arjen Markus described techniques for using the text widget for more than displaying text, highlighting the versatility of the text widget by using it to generate the PowerPoint-like slides for his presentation.

The text widget is one of the more sophisticated and powerful widgets in Tk. Not only can a script insert text into the widget, using a variety of fonts, foreground and background colors, and event bindings connected to the text, a script can also insert images and other Tk widgets.

Markus pointed out that the ability to put other widgets onto the text widget provides another technique for geometry management which is particularly

well adapted to fill-out-a-form style applications. This makes it easy to automatically create a "Poor Man's GUI" from a description of the data in a form.

He also demonstrated an application of Knuth's "Literate Programming" paradigm by mixing text and code in the text widget. His example uses the text widget's bind command to trigger actions based on the text being inserted into the widget.

ORASTATE (ORACLE STRUCTURE TOOL AND TABLE EDITOR)

Martin Fickert

ORASTATE is an application that generates a graphical representation of Oracle tables from a database or user interaction. Martin Fickert described how this package allows the user to perform database queries or edit tables, fields, and relations via a simple GUI.

This application is designed to allow users to interact with an Oracle database without needing to learn SQL and database commands.

This application represents a table as a drop-down list in which the top element is the name of the table and lower elements are the names of fields. During default display, the list is rolled up, and only the table name is displayed. Upon receiving an <Enter> (cursor over) event, the label "unrolls," displaying the table fields, which can be queried and modified via pop-up dialog windows. Relations between fields are displayed with lines, and can be added by the user drawing lines in the display.

Database queries can be created and evaluated via a set of pop-up windows that allow the user to select fields, operations between fields, and values for the operations.

UIE: CREATING A GENERIC INTERFACE BETWEEN TCL/TK AND OTHER LANGUAGES

Veronica Loell

Veronica Loell described the User Interface Engine (UIE), which uses Tcl as an

engine rather than as glue. This framework supports multiple software agents, cooperating to provide a natural language interface with other applications.

The UIE framework has several cooperating processes, including a data manager, translators, and applications which operate by passing messages to each other via the translators.

The translation engines allow different applications to communicate cleanly by converting application-specific commands into a generic intermediate language and back.

TCL USED FOR TESTING IN THE MOBILE WORLD

Hans Banken

SIGOS GmbH uses a Tcl application to test mobile networks. Hans Banken described how the application is constructed and how it works.

The requirement was to simulate user behavior using real mobile equipment. The application needed to be able to exercise all the functions a user could request, utilize all available hardware, and be easily configurable.

Banken described the SITE system, a database-backed test engine capable of running 180 distributed test systems. This system uses Tcl for the GUI, glue, and database interactions. The tests are described in a Tcl-style configuration file that allows the Test Definition Language to be easily converted to Tcl scripts for execution.

Banken noted that with no previous Tcl experience the team was able to construct a large, robust, and successful commercial application in only two weeks.

DBWIDGET: A RAPID APPLICATION DEVELOPMENT SYSTEM FOR THE TCL COMMUNITY

Franco Violi

Franco Violi described the dbWidget package developed to easily port legacy COBOL applications to modern envi-

ronments. The name dbWidget is a misnomer, given that the package is more than an interface to databases, and more than a widget. This package allows a developer to define an application with a template file that is used to generate the GUI and database interactions. The applications inherit base functionality from the dbWidget package, including audio cues, online help, and support for printing.

The dbWidget package interfaces with Open Office to provide platform-independent printing, and uses a C database extension to interface to the Berkeley db, Faircom's Ctree, or other database engines.

This package was developed to upgrade their clients' applications. It's been in use for several years and is robust and effective.

XOTCLIDE

Artur Trzewik

Artur Trzewik described XOTclIDE, an integrated development environment for XOTcl (Extended Objective Tcl). This IDE has a strong small-talk orientation and includes support for building components and version control.

Trzewik introduced XOTcl to the audience and noted that XOTcl uses a method-chaining style of inheritance and provides more support for introspection than [incr Tcl]. XOTcl also supports assertions, Mixins/Filters, and classes that contain metadata.

The XOTclIDE application allows the user to build an application in an interactive manner, constructing classes and methods bit by bit and testing them in a bottom up manner until the final top-level class (the application) is completed.

The XOTclIDE debugger is based on the atkdebugger, which allows the user to set breakpoints and step into/over procedures and modify the original source,

making the edit/test/debug sequence more integrated.

The Version Control is based on a relational database (MySQL, Postgres, SQLite, ODBC) that works in the background, rather than requiring explicit check-in/check-out sequences. This paradigm supports a team-programming style, allowing a user to check at any time, via a code browser, to find out what modules are in sync with the repository.

I18N: TCL FOR THE WORLD

Richard Suchenwirth

Richard Suchenwirth described Tcl's support for internationalization and localization of applications. He explained the evolution of Unicode, from the early Baudot code, through ASCII, EBCDIC, JIS, and UTF-8, and, finally, to the modern four-byte Unicode that can represent all known glyphs.

Suchenwirth demonstrated Tcl/Tk's facility with non-European fonts with a display from his iPAQ that showed similar phrases in several languages, including Chinese, Japanese, Korean, Greek, and Russian.

Since version 8.1, Tcl/Tk has used UTF-8, with support for up to three bytes per character, for internal character representation. Thus, a character may be a single or multiple bytes in the internal representation, but to a Tcl script each character is a single entity. He described using the msgcat package to support code localization and also how the text widget will automatically load fonts as required to display various glyphs.

He finished by discussing techniques for entering non-ASCII data from a standard keyboard. Combinations of shift and a letter can be mapped to other glyphs, or one could type the name of the character until the application recognizes the pattern and provides a menu of possible glyphs. Another option is to use a variant on greeklish, a style in which two character combinations are

used to represent non-ASCII characters. For example, *a'* would be used to represent the accented a character.

GENERATION R

Pascal Scheffers

Pascal Scheffers reported on the uses of Tcl by the Generation R project at the Erasmus University Medical Center in Rotterdam. Five university departments and over 20 external organizations are working together to track the lives of 10,000 children from pre-birth to young adulthood. This task involves many questionnaires and a great deal of information in many different databases maintained by the different groups working on the project.

The project is using AOLServer and the OpenACS package for much of the logistics and data management services. AOLServer is an HTTP server that uses Tcl as the scripting language for server-side processing. OpenACS is a package of tools for creating database-backed Web sites using AOLServer.

AOLServer provides a platform for building easy-to-use interfaces between the non-computer scientists using the data and the different database engines and data formats.

One continuing problem that the group faces is the need to generate customized form letters. After trying LaTeX, PageMaker, and MS-Word, they decided the optimal solution was to save the basic form letter as a PostScript file and then massage the text with a Tcl script. This solution provides the fastest mail-merge and print options.

Tcl's string and list manipulation features are also used to parse a set of Health Level 7 data. The ISO Level 7 data format is a sequence of character-separated, nested lists with different separation characters for each level of nesting. The Tcl split command makes parsing this data easy. The commercial parsing packages cost about \$6500.

Developing a Tcl solution from scratch took two days.

Scheffers finished his talk by noting that while Tcl has easy support for many different database engines, the binary library support is not always easy to acquire. The assumption is that anyone who needs to use a database engine has the required libraries and C compiler for the platform they are using. In a multi-disciplinary project like this, that's often not the case.

EMBEDDING THE TCL INTERPRETER FOR PORTABLE APPLICATIONS

Clif Flynt

The ease with which Tcl can be extended using other libraries and C code is well known and documented. Clif Flynt presented a tutorial talk describing how to embed the Tcl interpreter within a C application.

He described the Tcl library API that supports this, and demonstrated a short C/Tcl application that uses a C language main() function for the primary entry point and calculation function, and Tcl scripts for the GUI and data representation.

Flynt noted that this provides a "best of all worlds" development environment. The mission-critical, stable, platform-independent code is maintained in C, while the customer-experience, most-likely-to-be-changed, platform-dependent code is written in Tcl, where it can easily be modified to fit customer requirements. Because the user interface code is written in Tcl, it takes advantage of Tcl's multi-platform support, and what would otherwise be platform dependent becomes platform independent as well.

This ability to embed the interpreter makes the Tcl/Tk libraries a good alternative to using the Xt, Athena, or MS libraries for project development.