



EUROPEAN TCL/TK 2008 USER MEETING

Strasbourg (France)
6-7 June

<http://www.eurotcl.org/>



Abstract Book



Friday, June 6th

Session I – Tcl/Tk Anywhere (Jean Muller)

- Alexios ZAVRAS, *Tcl on the OLPC* 4
- Jos JASPERS, *VCRI: New Developments* 5
- Richard SUCHENWIRTH, *Tcl on the Phone* 6

Session II – Tcl core – Tk Image (David Zolli)

- Daniel STEFFEN, *Profiling and Debugging Tcl with Dtrace* 8
- Jean-Claude WIPPLER, *Why arrays (and dict) are great, yet totally inadequate* 9
- Arnulf WIEDEMANN, *BOF - tclOO, itclng and the Current OO* 10
- Jan NIJTMANS, *TkImg - Past, Present, Future* 11
- Paul OBERMEIER, *EMIT – The Extensible Multispectral Image Generation Toolset* 12
- Arnulf WIEDEMANN, *NexTk/nTkWidget – A possible Replacement for Tk* 13

Saturday, June 7th

Session I – Tcl/Tk and Science (Luc Moulinier)

- Raymond RIPP, *Tcl/Tk and Bioinformatics* 15
- Detlef GROTH, *Nfickle: Improved Scanner Generators for Tcl* 16
- Martyn SMITH, *Tcl/Tk's Contribution to Satellite Operations* 17
- Jan NIJTMANS, *SECS with Tcl* 18
- Daniel STEFFEN, *BOF - MacTcl/AquaTk* 19

Session II – Applied Tcl/Tk (Gérard Sookahet)

- Jean-Claude WIPPLER, *Mavrig - A Tcl Application Construction Set* 20
- Arnaud LAPREVOTE, *Ucome a New Tcl Web Environment* 21
- Daniel A. STEFFEN, *Tcl/Tk Community: Google Summer of Code* 22
- Franz WIRL, *GSoc Project: A Business Rule Management System based on the high-level object oriented scripting language XOTcl* 23

Session I - Tcl/Tk Anywhere

- Alexios ZAVRAS, *Tcl on the OLPC*
- Jos JASPERS, *VCRI: New Developments*
- Richard SUCHENWIRTH, *Tcl on the Phone*

Alexios ZAVRAS

Independent IT consultant, Greece

Keywords: Tcl, OLPC

The [OLPC](#) is an innovative computer which was designed by the team led by professor Nicholas Negroponte. The computer, also known as "children's laptop" or "\$100 laptop" is distributed on a not-for-profit basis to states and organizations to be distributed to children all around the world. Despite its small cost, the OLPC incorporates technological innovations, in both its hardware and its software. Its fundamental design principle is that it constitutes an educational tool for the children, and thus is not a simplified conventional computer.

Although the de-facto programming environment on the OLPC is Python, Tcl/Tk was ported on the OLPC and used for the creation of simple educational toys and demos. The port was straightforward, due to the Linux infrastructure of the OLPC software; however, the user interface used on the OLPC (specially designed for children) challenged some basic assumptions.

During the presentation, the OLPC machine will be presented and demonstrated (in both hardware and software), together with basic Tcl/Tk activities. Special attention will be given to some interface details implemented in other OLPC software, which might be useful for Tcl/Tk in the future.

Jos JASPERS

Utrecht University, Netherlands

Keywords: Tcl, education, synchronisation, client-server

In the PRO-ICT project a groupware environment has been created, to facilitate collaborative writing in project based learning assignments in secondary schools. [VCRI](#) (Virtual Collaborative Research Institute) is a tcl-based system that supports such activities. The base environment of VCRI included three tools: a database of relevant information sources (Database), a chat facility for deliberation between the students (Chat) and a shared text processor (Co-writer). The Co-writer allows the students to work on the texts of their assignments simultaneously. The system can be used synchronously and a-synchronously.

Recently the system has been expanded with a simulation/modeling tool and collaboration evaluation tool. The presentation will focus on the implementation of the new tools, a short demo and some strategic issues.

Richard SUCHENWIRTH

Siemens I MO IL RC, Germany

Keywords: eTcl, PDA, cellphone, IDE

Experiences with various Tcl ports on PDAs are reported. [ETcl](#) is considered the most actively developed port for Windows CE at current, and has proved to be satisfying and usable.

In contrast to earlier monolithic ("cathedral") approaches, a "patchwork bazaar" of small and independent extensions was developed that extend the eTcl console with useful tools for visualisation etc. The code is available freely on the Tclders' Wiki, under [Sepp](#).

Session II – Tcl Core – Tk Image

- Daniel STEFFEN, *Profiling and Debugging Tcl with Dtrace*
- Jean-Claude WIPPLER, *Why arrays (and dict) are great, yet totally inadequate*
- Arnulf WIEDEMANN, *BOF - tclOO, itclng and the Current OO*
- Jan NIJTMANS, *TkImg - Past, Present, Future*
- Paul OBERMEIER, *EMIT – The Extensible Multispectral Image Generation Toolset*
- Arnulf WIEDEMANN, *NexTk/nTkWidget – A possible Replacement for Tk*

Profiling and Debugging Tcl with Dtrace

Daniel STEFFEN

Independant, Switzerland

Keywords: profiling, tracing, DTrace, Tcl core

DTrace is an extensible dynamic tracing framework created by Sun, it is available in Solaris 10, Mac OS X Leopard and FreeBSD. Support for tracing Tcl programs with DTrace was added to the Tcl core in 8.4.16 and 8.5b1 by the speaker (c.f. <http://wiki.tcl.tk/DTrace>).

After an introduction to DTrace and the D programming language, the talk will present the new probes made available by the Tcl DTrace provider and illustrate how they can be used in conjunction with the existing system-wide probes to profile and debug Tcl programs without disturbing the program when tracing is not in use; demonstrations will show DTracing of Tcl in action on Mac OS X and Solaris.

Why arrays (and dict) are great, yet totally inadequate

Jean-Claude WIPPLER

Equi 4 Software, Netherlands

Keywords: Tcl, data structures, persistence

There is a dichotomy between creating Tcl applications (files, packages, databases) and running them (vars, arrays, dicts). The former is about designing data structures and code to deal with them, the latter is about handling actual data. We are used to "loading" and "saving" data as transition between these two phases, using files and/or databases. Although Tcl excels as load/save since "everything is a string", that doesn't really scale for non-trivial datasets.

This presentation examines various options and trade-offs in the context of Tcl, where code itself is also data. Some examples will illustrate how files, databases, dicts, namespaces, scripts, packages, and extensions all try to address similar issues yet with a considerable level of impedance mismatching. There is surprisingly little reuse of concepts and code when it comes to traversing tree structures (file system, namespaces, xml) or tabular data (lists, dicts, arrays, database tables, dir listings).

It will be shown that by adding thin wrappers around existing techniques and tools, a lot of generality could be gained in Tcl with little effort.

BOF - tclOO, itclng and the Current OO

Arnulf WIEDEMANN

Siemens SIS GO GAA GPCG BI, Germany

Keywords: OO

Jan NIJTMANS

Siemens I MO IL RC, Germany

Keywords: Tklmg

Paul OBERMEIER

MBDA Missile Systems, Germany

Keywords: Tcl, Testing, Wrapping, Rendering

EMIT is a set of libraries and tools to generate synthetic images in the visual and infrared spectrum. It is used at MBDA-Germany to support development and validation of existing and future missile systems.

Tcl/Tk is used to expose the functionality of the C/C++ based libraries at the scripting level for testing purposes and implementation of platform independent user interfaces.

The presentation will give an overview of the EMIT architecture and show some examples of Tcl/Tk usage in the toolset. An in-depth view of the wrapping techniques will complete the presentation.

NexTk/ntkWidget - a possible replacement for Tk

Arnulf WIEDEMANN

Siemens SIS GO GAA GPCG BI, Germany

Keywords: Tk, NexTk, ntk, widget, graphic

The current implementation of Tk has a lot of leftovers of the long history it has, including a lot of dependencies from X11 code which it calls often directly.

Therefor and for other reasons George Peter Staplin started some years ago an implementation of Tk9 later on renamed to NexTk together with megapkg, ntk ad freetypeext.

The basic idea behind is using images for the widget and also for all the parts of a widget.

Ntk widget is based on that implementation with some differences and with the additional use of GLFW a Graphic Library Frame Work and using openGL as the library for displaying images with the same interfaces used for linux, windows and macOS.

Session I – Tcl/Tk and Science

- Raymond RIPP, *Tcl/Tk and Bioinformatics*
- Detlef GROTH, *Nfickle: Improved Scanner Generators for Tcl*
- Martyn SMITH, *Tcl/Tk's Contribution to Satellite Operations*
- Jan NIJTMANS, *SECS with Tcl*
- Daniel STEFFEN, BOF - *MacTcl/AquaTk*

Raymond RIPP

IGBMC France

Keywords: Tcl, Biology, Bioinformatics

Research in Biology is no more possible without intensive use of computers and databases.

In our Laboratory of Integrative BioInformatic and Genomics (LBGI), we developed tools in Tcl/Tk allowing the analysis and management of a huge amount of heterogeneous data. This data is produced locally or by numerous worldwide bioinformatic centers. It covers various types of data such as biological sequences, from single protein or mRNA to whole genomes (3.4 GigaBases for the human genome), together with their complex features, as well as sets of experimental results or data treatments like transcriptomics data, comparison searches, multiples alignments, image processing, etc. The different types of data are stored in flat files, html pages or more or less well structured databases. Altogether this required the development of high throughput data retrieval, analysis procedures and data mining tools coupled to graphical interfaces and displays, and to specific databases and websites.

After a brief description of this context, we will present our Tcl/Tk developments made since 10 years by around 15 people, including biologists and students with very often minor background in programming languages. Tcl is easy to learn and allows quick progress.

Detlef GROTH

Potsdam University, Bioinformatics Group, c/o Max Planck Institute of Molecular Plant Physiology, Germany

Keywords: Tcl, scanner, flex, fickler, bioinformatics

In today's high-throughput world, (computational) biologists frequently face having to scan large amounts of data -- also called "parsing". Writing even short programs for these tasks is often tedious and usually error-prone. By utilizing C-based scanner generators like Lex and Flex, this work can be greatly simplified. Instead of having to write the complete programming code by hand, the scanner generator parses a file containing regular expressions and actions belonging to those expressions, thereby generating the entire source code for the entire application. From these input file, the source code for the application is generated. It has been shown that such automatically generated scanners are faster and easier to maintain than handwritten ones. Recently the author introduced a Tcl-based scanner generator called Ifickle which can be used to scan data with Itcl classes. Ifickle was a port of the Fickle scanner generator by Jason Tang. Whereas Fickle generates standalone executable Tcl-scripts, Ifickle generates Itcl- classes. However, other object-oriented frameworks like Otcl, XOtcl, and Snit are not supported. Because Fickle and its successor Ifickle are using the standard text-parsing approach for scanning flex-like input files, porting the scanner to other oo-frameworks and extending the scanner generator has been shown to be difficult. As a compiler should be able to compile itself, a scanner generator should be able to generate itself from a scanner declaration, i.e., a flex-like input file. This makes the addition of more features much easier.

In my talk I will present Nficle ("next fickler"), which supports scanners kept either in its own namespace or in object classes of various oo-frameworks. Some features like <<EOF>> actions and initialization patterns have been added as well. I will also compare scanner speeds and discuss options to improve the speed by adding C- code using Critcl.

Tcl/Tk's Contribution to Satellite Operations

Martyn SMITH

Spacebel, France

Keywords: Tcl, satellite, telemetry, simulation

The Space industry generates a lot of data, this data has to be processed, checked, analyzed.

I have been working for the CNES as a subcontractor for over 8 years, and have developed a number of tools in TCL/TK which are in daily use for processing satellite data.

PrestoDecom extracts telemetry data sent by the satellite in binary form into sets of text files for processing. PrestoPlot uses the BLT extension to graphically display timed data, it is also used as a batch tool to generate regular status reports.

CheckTM is a smaller tool to automatically apply a set of rules to validate telemetry data.

The simple and powerful nature of the TCL/TK combination and agile programming techniques resulted in the rapid production of fully functional tools.

Jan NIJTMANS

Siemens I MO IL RC, Germany

Keywords: Tcl, MES (Manufacturing Execution Systems), SECS (Semiconductor Equipment Communications Protocol), SEMI (Semiconductor Equipment and Materials International)

In the semiconductor industry, a widely used protocol is named SECS (Semiconductor Equipment Communication Standard). This protocol is used to connect the machines on the work floor with the rest of the automation. We will show how easy it is to implement a serial protocol (through socket or rs-232) like SECS with standard Tcl commands only.

A small framework is written to translate SECS messages to Tcl lists and back, and to open and maintain a connection to SECS-enabled equipment, all in a few hundreds lines of code. A short introduction will be given to the SECS protocol, and how knowledge of this protocol can directly be translated to implement a Tcl framework to handle that.

Finally we will give a demonstration of a (simulated) equipment that is controlled by this Tcl-SECS framework.

Session II – Applied Tcl/Tk

- Jean-Claude WIPPLER, *Mavrig - A Tcl Application Construction Set*
- Arnaud LAPREVOTE, *Ucome a New Tcl Web Environment*
- Alexios ZAVRAS, BOF - *The "One Laptop Per Child" demonstration*
- *Google Summer of Code + 1 Student presentation*

Mavrig - A Tcl Application Construction Set

Jean-Claude WIPPLER

Equi 4 Software, Netherlands

Keywords: Tcl

Rig and Mavrig are a set of tools to incrementally build applications in Tcl, whether command-line, network services, with a Tk GUI, or web-based. The generic Rig.tcl file provides a simple starting point for creating and growing highly modular and loosely-coupled applications. Rig includes a module loader, an event hook mechanism, logging and assertion facilities, and some utility functions. But the main purpose of Rig is to act as a set of non-intrusive conventions which help structure applications and the extensions they need. Mavrig adds a (still-evolving) set of modules and flexible data-structures to Rig to support application development further still, including network services, template-based web development, and Metakit-style data structures with relational operators and more.

The current state of the Mavrig project will be presented, along with a range of examples.

Ucome a New Tcl Web Environment

Arnaud LAPREVOTE

Mandriva/Edge-IT/Linbox, France

Keywords: Web, framework, template

Ucome is an environment for programming web applications in Tcl. It is not a web server, but an application development environment and also a new way to think application development.

Basically a web site is always a collection of file. Most of the time, these files are displayable by a web browser. Very unlikely, this is not the case of the normal content of user directories. We have files with many different formats. So creating a web site is never trivial, as we have to transform the files in some "good" web browser displayable format. Ucome does handle this work in an automated way. A file is transformed (many times) from an original format to a target format depending on an action asked. The default action is "view". The target may be html, pdf, text, or something else. There are many possible actions : view, edit, copy, delete, show_properties, menu, tree, ...

The processing of a file is rarely unique. Most of the time, the file is changed in intermediate format till being displayed. One particular intermediate format is "composite". In this format, it is determined the page template that will be used, and for each block of this template a new pipeline of transformation (file, action, target) is done. It allows to have a tree menu, a menu bar, an action bar, ... on each page in an easy way.

Ucome was developped in the context of the *Equal and Enthroned European projects*. The first application was to have web contents automatically extracted, transformed and displayed on a web site and on screens embedded in buses. This system was successfully tested in Metz in 3 buses during 6 months.

Daniel A. STEFFEN

Keywords: Tcl/Tk, GSoC

Franz WIRL

University of Vienna

Keywords: Tcl/Tk, GsoC, XOTcl