

MEJORAS AL SCEPTRE. EL PAQUETE SCEPTRE-INENCO.

Diego Saravia, Dolores Alía de Saravia
Facultad de Ciencias Exactas – Universidad Nacional de Salta (U.N.Sa.)
INENCO (U.N.Sa – CONICET)
Fax: 54-387-4255489, Av. Bolivia 5150. 4400 Salta, Argentina
email: Diego.Saravia@gmail.com, dsa@unsa.edu.ar,
loli@unsa.edu.ar, Dolores.Alia@gmail.com

RESUMEN: Se presenta la reorganización del paquete Sceptre a los efectos de incorporar las Autotools, producir paquetería instalable en los diferentes sistemas operativos y arquitecturas, modificar algunos de sus programas para adaptarlo a diferentes compiladores Fortran y permitir su autodetección, introducir tests y mecanismos para comprobar su buen funcionamiento y modificar los scripts para su invocación generalizando su funcionamiento. Con esto Sceptre se acerca al cumplimiento de diversos estándares del proyecto GNU, en cuanto a distribución de software, con lo que resulta más fácil de usar e instalar.

Palabras clave: Sceptre, Simusol, GNU/Linux, Autotools, Sumapack, Software Libre.

INTRODUCCION

Sceptre (Novender, 1999) está siendo adaptado en el INENCO para que:

- quede construido como un paquete que respete los estándares de programación del proyecto GNU (Free Software Foundation, 2009),
- pueda ser incorporado a las diferentes distribuciones GNU/Linux (Wikipedia, 2010c) y otros sistemas como Windows usando Cygwin (Red Hat, 2008),
- permita una mejor integración con otros sistemas, en particular con Simusol (Alía, Saravia, y Saravia, 2003, 2009),
- disponga de una interfaz gráfica,
- permita la colaboración de terceras partes mediante las metodologías colaborativas de desarrollo habituales en los sistemas libres (Raymond, 1998),
- reconozca y opere con Sumapack (Saravia, 2010b), cuando este instalado,
- facilite su instalación y uso (Salas, 2007).

A tales efectos se ha desarrollado el paquete Sceptre-inenco, mejorando el código Fortran, incorporando tests, generando los Makefiles con las Autotools (Wikipedia, 2010b), ubicándolo en un repositorio subversion (Apache Software Foundation, 2010), generando paquetería como tar.gz, rpm, tbz2 y deb y permitiendo su uso mediante una interfaz gráfica basada en XUL (Wikipedia, 2010a, Strom, 2009, Mozilla Foundation, 2010). en forma conjunta con Simusol. En otros artículos se describen los trabajos para generar entrada para el sceptre en base a diagramas creados con Dia (Gnome, 2010).

ANTECEDENTES

Sceptre: “System for Circuit Evaluation and Prediction of Transient Radiation Effects” es un software para analizar y simular circuitos y redes de propósito general, lineales o no-lineales, con capacidad para determinar condiciones iniciales, transitorias y de estado estacionario: AC y DC. Usa un language de entrada y métodos de variable de estado. Fue originariamente desarrollado por IBM para las Fuerzas Armadas Norteamericanas; liberado y portado a sistemas tipo GNU/Linux por el Prof. Novender. Redactado principalmente en Fortran 77.

Simusol es un software de simulación de sistemas de aprovechamiento de energía solar. Facilita la descripción de diferentes circuitos físicos y su simulación numérica a partir de diagramas día que traduce a archivos aptos para el Sceptre.

Venimos utilizando Sceptre (Saravia y Saravia, 2000) con diversos compiladores de Fortran como f2c, g77 y f77. Pero las nuevas distribuciones GNU/Linux han dejado de incorporarlos y los compiladores disponibles, protestan ante algunas sentencias del código original. En particular algunos “warnings” señalados como “delete feature” hacen temer que en un futuro dejarán de ser interpretados. Por lo tanto es urgente adaptar los códigos. Esto fue comentado a Novender, a quien se le enviaron nuestros primeros cambios para eliminar cierto tipo de “warnings” o errores. Luego de un tiempo publicó una nueva versión, Sceptre-2009, con algunos cambios que no incorporan lo sugerido por nosotros, pero que eliminan una buena cantidad de “warnings”.

Para asegurar la subsistencia a largo plazo del Sceptre es necesario dotarlo de una comunidad de interesados y desarrolladores y eso requiere el uso de herramientas y prácticas que favorezcan la misma.

METODOLOGIAS Y HERRAMIENTAS

Se utilizan las autotools, el gcc, el gfortran, f2c, f77, LaTeX (Knuth,1978), Muse (Olson, 2009), Sumapack, y los sistemas de paquetería de diferentes distribuciones como dpkg, apt, rpm zypper, urpmi, uget, etc. (Wikipedia, 2010d).

RESULTADOS

Paquetería

Se separó el paquete Sceptre-inenco en 2 paquetes fuente: uno con el código y otro con la documentación. El paquete con código puede generar varios paquetes derivados para su instalación y distribución conteniendo binarios y software independiente de la arquitectura.

En el caso de Sceptre no es obvio aplicar la lógica de separación entre fuentes y binarios, porque para usarlo es necesario poseer el compilador de Fortran y linkear los código objeto generados, con los distribuidos por Sceptre.

Entonces se debe tener en la máquina en que se lo use todo el entorno de desarrollo que es necesario para empaquetarlo. Cada vez que cambia o se actualiza el compilador, -gfortran es parte de gcc (Free Software Foundation, 2010b)- se deben recrear los objetos previamente distribuidos.

Esto hace particularmente comprometida su incorporación a repositorios de diferentes distribuciones que deberán recompilar y reempaquetar Sceptre cada vez que se actualicen los compiladores.

Por otro lado no es conveniente, instalar un software compilado por uno mismo directamente en un sistema, porque pierde los beneficios que prestan los sistemas de manejo de paquetería como las herramientas apt, rpm y emerge.

Entonces la solución propuesta es que las personas instalen las fuentes y los binarios con las herramientas previstas para cada sistema y que con las fuentes puedan auto-generar paquetes binarios actualizados específicos para su compilador en forma simple e instalar localmente éstos, cuando los necesiten. Así, cada vez que actualizan su distribución, pueden obtener los binarios desde el repositorio de Sceptre-inenco o bien recompilar el Sceptre, generar los paquetes e instalarlos en su propia máquina con el sistema de paquetería habitual.

También se busca que la instalación del paquete binario sea suficiente para que los usuarios lo puedan usar directamente sin necesidad de ejecutar una instalación individual; para esto se ha mejorado la lógica de los scripts de uso.

Todo el software distribuido respeta la licencia libre original del paquete de Novender (Free Software Foundation, 2007) y es Software Libre (Free Software Foundation, 2010a).

Estructura del paquete fuente

Se cambió la estructura de directorios de las fuentes Sceptre para hacerla más parecida a la usada habitualmente en los proyectos con Autotools. Los directorios usados son: data, examples, m4, scripts, test, y src. Se crearon los archivos configure.ac, y los Makefile.am en todos los directorios. En:

- data se colocan los archivos ngp.rc, modelib.dat y userlib.a,
- examples, todos los ejemplo, incluyendo cada uno sus archivos con resultados para validar las pruebas,
- test el script para realizar todas las pruebas,
- scripts, los archivos que llaman al Sceptre o que lo recompilan, también el ngp,
- m4 los archivos respectivos del autoconf,
- src los tres directorios x1 x2 y x3 renombrados como phase1, phase2 y auxpro.

Generación de paquetes binarios

Se proveen comandos en los Makefiles para generar paquetes deb, rpm, tgz y tbz2.

Se utiliza la infraestructura de Sumapack para esto. Se realizan diferentes paquetes para diferentes procesadores y sistemas operativos.

Para instalar Sceptre desde los repositorios se debe hacer un svn checkout desde:

<http://www.inenco.net/repos//loli/sceptre-inenco/trunk>
<http://www.inenco.net/repos/dsatex/sceptre-inenco-doc/>

y luego proceder con los pasos habituales.

```
svn co http://www.inenco.net/repos//loli/sceptre-inenco/trunk
cd trunk
autoreconf
automake --add-missing
./configure
```

```
make
make DESTDIR=install install
make binpackages
sudo dpkg -i sceptre-inenco-xxx.deb (ejemplo para debian o ubuntu)
sudo rpm -i sceptre-inenco-xxx.rpm (fedora u opensuse)
```

El objetivo binpackages es provisto por Sumapack
Sumapack también puede ayudar a instalar las dependencias de Sceptre.
Si no esta instalado Sumapack, el paquete sceptre igual puede producir binarios e instalarse directamente.

Manejo de los paquetes de documentación

Se ha creado el paquete Sceptre-inenco-doc que contiene varios documentos relativos al Sceptre y al Simusol, preparados con Muse o bien en pdf.

Como estos documentos requieren la instalación de software muy diferente al paquete original, como LaTeX, se ha preferido mantenerlo separado para simplificar la instalación del paquete básico en máquinas que no requieran compilar documentos.

Paquete con scripts ejecutables independientes de la arquitectura

Básicamente un script para compilar y ejecutar el Sceptre sobre cada instancia de uso y otro para recompilar el Sceptre si es necesario.

Cambios para su uso por múltiples compiladores Fortran

El Fortran 77 admitía algunas modos de programar que fueron muy útiles en épocas anteriores cuando los recursos de memoria eran escasos, pero que ciertamente eran complicados de entender; y mal utilizados, eran causantes de errores difíciles de detectar. Los compiladores Fortran disponibles actualmente no aceptan bien esos modos: Algunos los aceptan produciendo avisos, “warnings”, alertando sobre un uso no conveniente y otros simplemente abortan la compilación con mensajes de error.

En la utilización del Sceptre hay dos tipos de compilaciones:

- La inicial, al momento de la instalación, en la que se compilan todas las fuentes y se prepara un ejecutable que traducirá el archivo .d de entrada para el Sceptre a un archivo .f que provocará la simulación.
- Las “particulares”: correspondientes a cada uno de los archivos creados por Sceptre como traducción de algún archivo .d. Aquí la variedad de compilaciones posibles es infinita.

Ha sido un objetivo de los cambios introducidos a las fuentes Fortran de Sceptre eliminar todos los warnings de las compilaciones. El objetivo se ha cumplido, para los compiladores g77 (gcc 3.3.5) y gfortran (gcc 4.3.1), tanto para su compilación inicial como para todas las compilaciones “particulares” que se han probado. No se puede asegurar que no haya novedades en ese sentido; será necesario seguir actuando.

Ciertamente los warnings producidos al compilar código Fortran ayudan a mejorarlo: los avisos de etiquetas o variables no utilizadas han sido fáciles de tener en cuenta; algunos de los avisos de argumentos de subrutinas no utilizados también; pero otros han sido más difíciles por cuanto algunas de las subrutinas definitivas serán escritas por el propio sceptre ante una instancia de uso. Al compilar Sceptre-2009 con los compiladores nombrados se producen muchos menos warnings que al compilar Sceptre-2006. En nuestro trabajo hemos aprovechado todas las correcciones efectuadas en el Sceptre-2009.

Cambios para simplificar y estandarizar su uso

Hasta la versión del 2009, todas los programas Fortran del Sceptre trabajan siempre con los mismos archivos de entrada y salida: hay un script que hace la copia del archivo de entrada provisto por el usuario a un input.dat y luego copia los archivos de salida a archivos con nombre propio igual a la parte propia que el archivo .d. (diferentes extensiones). Hay archivos intermedios que no son cambiados, en especial el ejecutable sceptre2 (el simulador). Esto permite tener código Fortran más sencillo.

Hemos creído conveniente abandonar esa práctica; ello ha facilitado el estudiar el funcionamiento del Sceptre, lo cual ha sido imprescindible para atender a los warnings relativos a algunas de las compilaciones particulares: avisos acerca de tamaños diferentes para un mismo “common”, arrays declarados de tamaño 1 que se usan con índices mayores. Otros cambios introducidos a efectos de facilitar la comprensión del código han consistido en conseguir que tanto las variables de los “common”, como las unidades lógicas de entrada-salida tengan el mismo identificador en todas las rutinas. Todo ello ha permitido detectar, por ejemplo, que los resultados de un transitorio que el Sceptre guarda en un archivo binario dirac.dat pueden volcarse directamente a un archivo de texto haciendo innecesaria su traducción posterior; con la ventaja adicional de que este archivo de texto no tiene limitación prevista en cuanto a su tamaño mientras que el dirac.dat sí la tiene. Esto permitía suprimir la producción del archivo dirac.dat y del programa que hace la traducción posterior con ngp (ngp_gnu.f); no se ha hecho para no modificar las condiciones de uso del Sceptre.

Pruebas sobre el funcionamiento

Es parte de los modernos procesos de desarrollo de software el utilizar sistemas para comprobar el funcionamiento de los programas (Free Software Foundation, 2008). En el caso de Sceptre y con los cambios requeridos por los diferentes compiladores, notamos que en algunos casos todo aparentaba funcionar correctamente, pero en realidad, el sistema no calculaba correctamente. Por lo tanto se hizo imprescindible tener a mano un conjunto de pruebas automatizadas que nos permitan asegurar que ante un cambio, todo lo importante seguía funcionando correctamente.

Durante el desarrollo, se han hecho muchas pruebas para estudiar la influencia del compilador Fortran y los cambios introducidos en los programas Fortran. Se ha incorporado al paquete la posibilidad de probar el Sceptre-inenco ya instalado comparándolo con los resultados obtenidos en el lugar de desarrollo; estos resultados han sido incorporados al paquete. Naturalmente las pruebas podrán ser usadas con propósitos similares.

Cada prueba consiste en ejecutar un archivo .d y detectar si los archivos de salida son “iguales” o no a los guardados. Escribimos iguales entre comillas porque se descartan las diferencias provenientes de la fecha de ejecución, o de la duración de algún cálculo, o cambios de notación aceptables. Se han tomado todos los archivos que vienen como ejemplos en las versiones clásicas del Sceptre y algunos otros.

Interacción de Simusol con Sceptre

Simusol básicamente produce archivos .d para el Sceptre a partir de los archivos creados por el usuario utilizando Dia. Simusol le pide casi siempre al Sceptre que calcule transitorios; luego toma los resultados del Sceptre, los interpreta y los prepara para su visualización mediante el graficador Gnuplot. Todo ello en el marco de un archivo Makefile que puede ser controlado manualmente o por una interfaz gráfica basada en XUL que se distribuye dentro del paquete Simusol-Inenco.

Cada proyecto debe estar contenido en un directorio, cuyo nombre también se usa para los archivos del mismo. La interfaz gráfica selecciona el directorio (proyecto). En ese momento si no existe, crea un Makefile a partir de un template. Este Makefile interactúa correctamente con Dia, Sceptre y Simusol y es usado por la interfaz, aunque se puede manipular también manualmente.

CONCLUSIONES

Se ha creado un “fork” del Sceptre para avanzar con los cambios que necesitamos y que eventualmente estas contribuciones puedan ser tomadas por el proyecto original si lo considera adecuado.

Se invita a todos los interesados a contribuir mejoras y participar en el desarrollo.

El paquete Sceptre-inenco es más fácil de usar, auditar, distribuir y está más en línea con los estándares de calidad del mundo GNU/Linux. Los cambios realizados también facilitan su uso en Windows.

De esta forma, y también con el proyecto Simusol, ayudamos a que Sceptre continúe vivo y se mantenga vigente en el tiempo, ya que es un software excepcionalmente útil y simple para estas tareas.

REFERENCIAS

Alía de Saravia, D., Saravia, L. y Saravia, D. (2003). Simusol: Simulating thermal systems using Sceptre and Dia
http://www.frcu.utn.edu.ar/deptos/depto_3/32JAIIO/jsl/JSL_01.pdf

Alía de Saravia, D., Saravia, L. y Saravia, D. (2009). Simusol
<http://www.simusol.org.ar>

Apache Software Foundation, The (2010). Apache Subversion
<http://subversion.apache.org/>

Free Software Foundation (2007). Licencia AGPL. Ver también la GPL.
<http://www.gnu.org/licenses/agpl.html>

Dejagnu
<http://www.gnu.org/software/dejagnu/>

Free Software Foundation (2009). GNU Coding Standards
<http://www.gnu.org/prep/standards/standards.html>

Free Software Foundation (2010a). Software Libre
<http://www.gnu.org/philosophy/free-sw.es.html>

- Free Software Foundation (2010b). GCC, the GNU Compiler Collection
<http://gcc.gnu.org/>
- Gnome (2010). Dia
<http://live.gnome.org/Dia>
http://dia-installer.de/index_en.html
- Knuth, D. (1978). TeX
<http://www.tug.org>
<http://es.wikipedia.org/wiki/TeX>
LaTeX: <http://es.wikipedia.org/wiki/LaTeX>
- Mozilla Foundation (2010). XUL. Mozilla Development Center
<https://developer.mozilla.org/En/XUL>
- Novender, W. (1999). Sceptre: Simulation of Nonlinear Electrical Circuits Issue 63. July.
<http://www.linuxjournal.com/article/3008>
<http://alpha.fh-friedberg.de/iem/84.html>
http://alpha.fh-friedberg.de/iem/fileadmin/user_upload/novender/sceptre/sceptre-2009.225.tar.gz
- Olson, M. (2009). Emacs Muse.
<http://mwolson.org/projects/EmacsMuse.html>
- Proyecto GNU (2010). Gfortran — the GNU Fortran compiler, part of GCC
<http://gcc.gnu.org/wiki/GFortran>
- Raymond, E S. (1998). La Catedral y el Bazar
<http://www.sindominio.net/biblioweb-old/telematica/catedral.html>
- Red Hat (2008). Cygwin.
<http://www.cygwin.com>
- Salas, G. (2007). Instalar Simusol: gcc y g77 en Ubuntu 8.10
<http://ubuntuforums.org/showthread.php?t=1147530>
- Saravia, D. y Saravia, L. (2000). Ututo GNU/Linux. Inenco UNSa, ISBN 987-9381-06-8. Disco Compacto. Simulando sistemas solares con sceptre.
<http://softwarelibre.unsa.edu.ar/ututo/ututo.html>
Actual: <http://www.ututo.org>
- Saravia, D. (2010b). SumaPack
<http://www.sumapack.org>
- Strom, E. (2009). XUL::Gui - render cross platform gui applications with firefox from Perl
<http://search.cpan.org/~asg/XUL-Gui-0.36/lib/XUL/Gui.pm>
- Wikipedia (2010a). XUL
<http://en.wikipedia.org/wiki/XUL>
- Wikipedia (2010b). GNU build system
http://es.wikipedia.org/wiki/GNU_build_system
- Wikipedia (2010c). Distribución Linux
http://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux
- Wikipedia (2010d). Sistema de gestión de paquetes.
http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_paquetes

ABSTRACT

This paper introduces the Sceptre-inenco package, a reorganization of Sceptre that: uses Autotools; produces deb, rpm and tbz2 packages to install natively in different GNU/Linux systems; modifies several fortran sources to compile under gfortran; adds a testing framework and several example based tests; and provides better scripts for running and recompiling Sceptre. With these changes Sceptre, aligns itself better with GNU standars, being more easy to install and distribute.

Keywords: Sceptre, Simusol, GNU/Linux, Autotools, Sumapack, Software Libre.