SAMOS: Desarrollo de un Sistema de Dinámica de Vuelo basado en Software Libre, para un Centro de Control de Satélites

Gonzalo García Julián (gggj@gmv.es), Ricardo Píriz Núñez (jrpn@gmv.es), Antonio Pérez Cambriles (anpc@gmv.es), José Luis López Martín (jllm@gmv.es), Manuel Fernández Fernández (maff@gmv.es), Juan Carlos Vallejo Chavarino (jcvc@gmv.es)*
GMV - c/ Isaac Newton 11, 28760 Tres Cantos, Madrid, España
http://www.gmv.com

11 de noviembre de 2000

Esta ponencia forma parte del III Congreso Hispalinux[1] virtual, celebrado en Noviembre de 2000. Puedes participar en él en la sección de eventos de BarraPunto[2]

Resumen

En el marco del programa **ARTEMIS** (*Advanced Relay TEchnology MISsion*), satélite geoestacionario de la Agencia Europea del Espacio, la Unidad de Centros de Control de la empresa española GMV desarrolló, validó e integró entre los años 1996 y 1999 el Sistema de Dinámica de Vuelo **SAMOS** siguiendo una aproximación en la que el uso de software libre fue una constante.

El objetivo fundamental de esta ponencia es compartir la experiencia del uso de software libre en un proyecto de desarrollo de software comercial en el sector espacio. Las conclusiones fueron plenamente satisfactorias y de hecho nuevos proyectos en GMV siguen utilizando Linux como plataforma de desarrollo así como diversas herramientas de software libre.

[1] `http://congreso.hispalinux.es`

[2] `http://barrapunto.com/eventos`

Figura 1: ARTEMIS

1 Introducción

ARTEMIS es un satélite geostacionario de la Agencia Eurpoea del Espacio que será puesto en órbita en la primera mitad del año 2001. En su desarrollo ha participado un gran número de empresas europeas. ARTEMIS es fundamentalmente un demostrador de nuevas tecnologías de comunicaciones. Su carga de pago avanzada permitirá entre otras cosas la comunicación entre satélites de baja órbita (< 1500 km de altitud) y estaciones de tierra, a través de ARTEMIS, usando tanto comunicaciones en banda K y S como haces ópticos (láser). El control de la órbita se realizará mediante una combinación de propulsión convencional (química) y propulsión iónica.

Entre los usuarios futuros de ARTEMIS se encuentran satélites muy representativos de diversas agencias espaciales, como ENVISAT (ESA - Europa), SPOT-4 (CNES - Francia), OICETS (NASDA - Japón) y probablemente la Estación Espacial Internacional.

La empresa **GMV** lleva más de 15 años ocupando un lugar destacado en programas del sector espacio en Europa, tanto públicos como privados. En la actualidad emplea a cerca de 400 personas y desarrolla actividades en áreas tan diversas como espacio, defensa, tecnologías de la información, telecomunicaciones y transporte. El uso de Linux como plataforma de desarrollo ha sido algo muy frecuente desde los primeros años de la vida del sistema operativo, mucho antes de que alcanzara la popularidad actual. Diversos sistemas a medida sobre Linux han sido desarrollados por GMV; entre ellos SAMOS ocupa un lugar muy destacado.

El sistema **SAMOS** (*Satellite Attitude Monitoring and Orbit control System*) creado por GMV (ver 1) será responsable de la determinación de la órbita de ARTEMIS a partir del "*tracking*" recibido (medidas proporcionadas por las estaciones de seguimiento de satélites), la generación de información asociada a la órbita (eventos operacionales como eclipses, ángulos de apuntamiento de antenas y otros), el cálculo de las maniobras necesarias para el control de la órbita, la monitorización de los parámetros de telemetría, la determinación de la actitud del satélite en tiempo real y algunas funciones asociadas a la carga de pago y a los sistema de a bordo usados para el control de actitud. SAMOS cubre todas estas funciones tanto durante la fase inicial tras el lanzamiento (LEOP - *Launch and Early Orbit Phase*) como durante la fase de rutina (GEO - *Geostationary Earth Orbit*).

Esta ponencia está estructurada de la siguiente forma:

- La sección 2 cuenta las características funcionales de SAMOS.
- La sección 3 describe los elementos de Software Libre que han sido usados en SAMOS.
- La sección 4 justifica la elección de elementos de Software Libre para SAMOS.
- La sección 5 comenta las lecciones aprendidas durante el desarrollo y validación de SAMOS, en relación con el uso de Software Libre.
- La sección 6 expone las tendencias en el ámbito de los Centros de Control de Satélites, en particular con respecto al uso de Software Libre.

2 Características de SAMOS

SAMOS está diseñado como un sistema de **dos capas** (*two-tier architecture*): interfaz de usuario y cálculo. El interfaz de usuario, desarrollado en Tcl / Tk, proporciona acceso a las ventanas de control de los múltiples procesos de cálculo, ventanas de edición de datos, visualización de resultados y ayuda on- line. La capa de cálculo está desarrollada en FORTRAN 77, con el objetivo de reutilizar determinados módulos de software que se encuentran operando otros satélites desde hace tiempo y son, por tanto, muy fiables. Un módulo de comunicaciones, escrito en C, intercambia información (en particular, recibe telemetría) con el núcleo del centro de control, via RPC (*Remote Procedure Call*).

Las figuras 2, 3, 4 y 5 muestran varias ventanas del interfaz de usuario.

SAMOS está diseñado como un sistema altamente **modular** de manera que se permita la actualización de partes del sistema y la integración de nuevos módulos de cálculo, tanto durante la vida operativa de ARTEMIS como en potenciales adaptaciones a otras misiones.

SAMOS puede operar en tres **entornos** diferentes: Operacional, Simulación / Entrenamiento y Mantenimiento. Esto permite que por ejemplo se niegue el acceso a módulos de simulación desde el entorno operacional. Múltiples usuarios pueden utilizar el sistema simultáneamente, desde distintas máquinas gracias a XFree.

El sistema da soporte a toda la vida del satélite, en particular proporcionando **funciones específicas** para las siguientes fases:

- LEOP: Fase de lanzamiento y órbita de transferencia hasta que se lleva al satélite a su órbita nominal geoestacionaria. En esta fase, SAMOS proporciona algunas funciones críticas, como la Optimización de las maniobras de transferencia

- GEO: Fase de rutina, en la que se controla al satélite dentro de su ventana (station keeping) en la órbita geoestacionaria y éste da los servicios proporcionados por su carga de pago. En esta fase el sistema da soporte a tareas como la planificación de las maniobras de control de la órbita, la calibración de los motores cohete o el apoyo a ciertas funciones de la carga de pago.

Igualmente, SAMOS proporciona **funciones genéricas** de utilidad durante ambas fases:

- Simulación de datos de tracking.

- Preprocesado de datos de tracking

- Determinación y propagación de la órbita del satélite

- Archivo y recuperación de datos de órbita

- Análisis del consumo de la masa de propulsante del satélite

- Generación de productos, como ángulos de apuntamiento de antenas terrestres, visibilidad de estaciones, eventos orbitales (eclipses, colinealidad, ...).

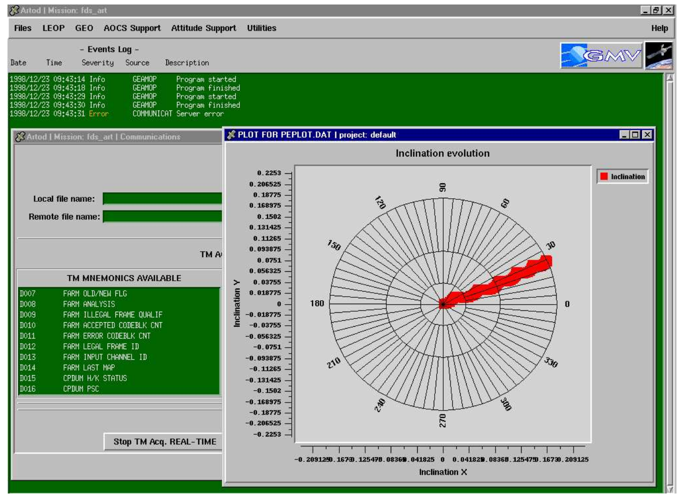

Figura 2: SAMOS

- Visualización de la secuencia de eventos orbitales (maniobras, eclipses, etc.), en forma de diagrama de Gantt (extensión añadida durante el mantenimiento).
- Soporte a algunos modos específicos de control de actitud
- Soporte a funciones del AOCS de a bordo (Attitude and Orbit Control Subsystem)
- Monitorización de la actitud del satélite en tiempo real y en play-back.

3 Software Libre usado en SAMOS

SAMOS corre en un PC bajo **SuSE Linux** (instalación inicial en el Centro de Control sobre la versión 5.3; diversas actualizaciones se han llevado a cabo desde entonces). Se utilizaron múltiples herramientas de software libre para el desarrollo, de GNU (g77, gcc, gdb, RCS, ...) así como de otras fuentes (ddd, NEdit, ...). El interfaz de usuario se desarrolló en Tcl/Tk y con extensiones como BLT. Algunos módulos específicos utilizados para realizar envíos de ficheros mediante FTP se escribieron en Expect.

El uso de **Tcl/Tk** en la Unidad de Centros de Control de GMV ha sido una constante desde antes del desarrollo de SAMOS. Su alta portabilidad, facilidad de aprendizaje y elevada productividad lo hacen una opción ideal para el tipo de desarrollos a medida que se han realizado para diversos clientes en los últimos años. De hecho, GMV ha desarrollado una herramienta, **tkForms**, para la generación rápida de paneles en Tcl/Tk, que permite

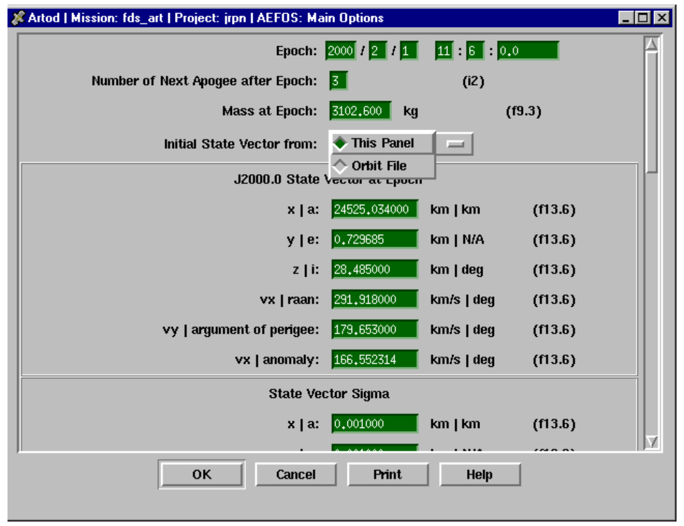

Figura 3: Ejemplo de un panel de edición

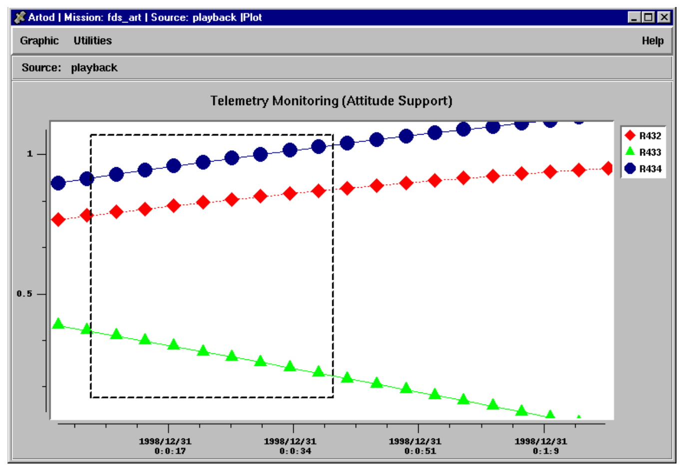

Figura 4: Monitorización de Telemetría

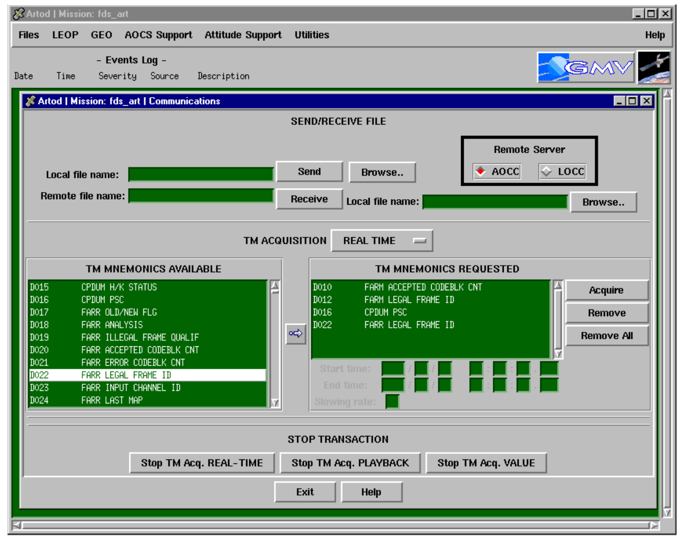

Figura 5: Consola de Comunicaciones

integrar rápidamente en un sistema (en particular, en SAMOS) un módulo de cálculo nuevo. Esta herramienta es tan fácil de usar que cualquier ingeniero de proyecto puede crear rápidamente el interfaz gráfico de un módulo nuevo sin tener conocimientos de Tcl/Tk. Este tipo de soluciones es esencial para sistemas como SAMOS, que tiene más de 200.000 líneas de código y agrupa unos 80 programas de cálculo diferentes.

4 ¿Por qué Linux y software libre?

El presupuesto total de SAMOS superó los 140 millones de pesetas, de las cuales un 100% corresponde a valor añadido por GMV (no se entregaron ni hardware ni licencias de software comercial). Ante este presupuesto queda claro que los argumentos para el uso de software libre no eran exclusivamente los ahorros de costes asociados. De hecho, las razones fundamentales para elegir esta vía fueron otras:

- **Acceso al código fuente:** Es la única manera de garantizar la posibilidad de realizar correcciones o mejoras sobre el sistema durante toda la vida operativa del satélite (más de 10 años). La historia demuestra que nunca se sabe cuando un suministrador de software va a dejar de soportar un producto o incluso va a desaparecer como empresa. Este riesgo es excesivo para sistemas que deben estar operativos durante más de una década.

- **Costes:** del software incorporado al sistema, del software de desarrollo y del hardware asociado, mucho menor para PCs que el derivado del uso de estaciones de trabajo UNIX. En el momento de tomar las decisiones de implementación del sistema, Windows NT apenas era usado en este tipo de centros. Desde entonces se ha revelado como una firme opción, sobre todo para usuarios que están acostumbrados a trabajar en dicho entorno con aplicaciones de ofimática.

- **Soporte:** La realidad demuestra que en muchos casos la comunidad internacional del mundo del software libre es capaz de proporcionar soluciones rápidas a problemas y mejoras de sus productos, de una forma mucho más eficiente que muchos suministradores de productos comerciales.

5 Lecciones aprendidas

La experiencia de GMV a lo largo del desarrollo, validación y primeros meses de mantenimiento de SAMOS ha confirmado como acertadas la elección de Linux como sistema operativo y el uso de software libre. Varios ejemplos pueden dar una idea de hasta qué punto esto ha sido así:

- En un momento crítico de la aceptación se detectó que el interfaz de usuario del sistema funcionaba cada vez más lentamente cuando había estado operativo durante un cierto tiempo, hasta llegar a niveles inaceptables de tiempo de reacción. Tras unos días de investigación, un ingeniero de GMV logró aislar y reproducir el problema en un breve script de Tcl / Tk. El problema se expuso en un **grupo de noticias** (newsgroup) de USENET, adjuntando el script, lo que dio lugar a una respuesta de otro experto en cuestión de horas. La respuesta identificaba el problema como una fuga de memoria (memory leak) bajo determinadas condiciones en la versión del intérprete de Tcl / Tk de Linux disponible en aquella fecha; no sólo eso: además proporcionaba una solución sencilla (corrigiendo una línea de un fichero de configuración y reinstalando el intérprete). Situaciones similares con productos comerciales han dado siempre lugar a costosos estudios para identificar y corregir el problema, y en algunos casos incluso han sido irresolubles.

- Recientemente Microsoft ha decidido dejar de vender licencias de **Windows NT**, obligando a adquirir Windows 2000 en su lugar. Este hecho va a tener un gran impacto sobre sistemas operacionales y en desarrollo sobre NT que puedan necesitrar licencias adicionales en un momento dado (se asegura que la migración es inmediata pero la realidad ya ha demostrado que surgen problemas). Decisiones como ésta, inesperadas y derivadas de argumentos comerciales de dicha empresa (que se aprovecha de su situación de monopolio) refuerzan los argumentos en favor del uso de software libre.

- La mayoría de los productos de software libre cumplen rigurosamente **estándares internacionales**. Por ejemplo, los compiladores de GNU soportan en la mayoría de los casos el 100% del estándar ANSI de cada lenguaje y permiten pocas extensiones. Esto ha sido una ventaja a la hora de portar código a otras plataformas, tarea que se ha realizado sin apenas problemas en todos los casos. Por contra sí que se encontraron múltiples problemas al intentar portar a Linux software proveniente de plataformas como VMS, por el uso de múltiples extensiones no estándar.

- No todo ha sido positivo. Un **inconveniente** claro es las limitaciones que presenta el depurador gdb para FORTRAN 77, claramente por debajo de compiladores existentes en otras plataformas (y de algunos disponibles en Linux, aunque éstos tienen un coste nada despreciable).

- Un aspecto a tener muy presente en productos comerciales es el de las **licencias** del software libre. Cada caso es diferente y es imprescindible estudiar con cuidado las restricciones existentes en relación con su uso comercial. En un desarrollo a medida es más fácil evitar esas restricciones dado que ni siquiera es necesario distribuir todo el software, siendo el cliente quien proporciona su propio entorno, que puede incluir elementos de software libre ya instalados.

6 Presente y futuro del Software Libre en el área de Centros de Control de Satélites

En opinión de los autores, las tendencias actuales en el sector Espacio en general y en el área de Centros de Control de Satélites en particular pueden resumirse en los siguientes puntos (véase por ejemplo 2):

- Presión cada vez mayor para **reducir los costes y los plazos** de desarrollo de sistemas, manteniendo o incluso aumentando los niveles de fiabilidad.

- En la línea del punto anterior se observa que cada vez es más habitual buscar soluciones basadas en productos comerciales genéricos o **COTS** (*Commercial Off-The-Shelf*). En cualquier caso existen muchos mitos acerca de estos COTS ya que en la mayor parte de los casos es necesario un esfuerzo considerable de configuración o incluso cambios mayores. Además es difícil dar soluciones genéricas en un sector de alta tecnología donde cada misión es diferente. También se comprueba que el nivel de soporte proporcionado suele ser insuficiente para las necesidades de los sistemas en este sector.

- Uso creciente de plataformas tipo **PC** en sustitución de estaciones de trabajo UNIX (SUN, HP, Compaq) o VMS, fundamentalmente con la intención de reducir costes de HW y SW. En estas plataformas se utiliza sobre todo Windows NT, la presencia de Linux es creciente pero todavía reducida.
- Sistemas distribuidos, comunicaciones fundamentalmente a través de **CORBA** (*Common Object Request Broker Architecture*), por haberse impuesto como estándar rápidamente a nivel mundial. De esta manera se intenta optimizar el uso de los recuros informáticos en el sistema y la integración entre componentes escritos en distintos lenguajes de programación y operando en plataformas heterogéneas. Implementaciones comerciales de CORBA, como Orbix, son las empleadas normalmente, de momento muy por encima de opciones en software libre como MICO.
- Programación en lenguajes altamente portables, sobre todo **Java**, y en menor medida, **Tcl/Tk** y **Perl/Tk**.
- Uso de estándares universales como **XML** para el intercambio de datos entre componentes, con la idea de lograr sistemas realmente abiertos.
- Potenciación del diseño orientado a objetos, usando nuevos estándares de notación como **UML** (*Unified Modelling Language*).

El uso de **software libre** en el sector espacial sigue siendo realmente escaso, como ha podido comprobarse fácilmente en eventos recientes como 2, en los que prácticamente se ha ignorado su existencia. Parece que siguen pesando mucho los mitos acerca de una supuesta menor fiabilidad y soporte en relación con productos comerciales. En cualquier caso su presencia en este sector es creciente, como lo demuestra la organización de seminarios específicos como 3. En el área particular de Centros de Control de satélites hay mayores posibilidades que en otras áreas como la de software embarcado (que impone unos requisitos muy estrictos y particulares). La presión sobre los costes ayudará sin duda a que su uso sea creciente en los próximos años, aunque no debe olvidarse que el coste del software adquirido es tan solo una parte del coste total del desarrollo de un sistema.

Empresas como **GMV** siguen apostando claramente por el software libre. Varios ejemplos pueden dar una idea de hasta qué punto:

- La **plataforma de desarrollo** individual estándar desde hace muchos años en la empresa es un PC con Linux (con la posibilidad de arrancar la máquina en modo Windows si se desea, gracias a Lilo). El programador puede conectarse a estaciones de trabajo potentes (vía X) o trabajar en modo local.

- El uso de **herramientas** de software libre para desarrollo está muy extendido (gcc, ddd, NEdit, source navigator, ...), incluso cuando se trabaja en plataformas comerciales (como SUN Solaris), salvo que éstas proporcionen alternativas mucho más potentes a un coste razonable. En muchos casos el coste de cada licencia de estos productos comerciales es excesivo si se tiene en cuenta las alternativas en software libre.

Referencias

1. *"SAMOS - A Flight Dynamics Full Mission Support System for ARTEMIS"*, 2nd International Symposium on Spacecraft Ground Control and Data Systems, Iguaçu, Brasil, February 1999, por M. A. Molina, J. Potti, G. di Genova, G. García-Julián, J. R. Píriz. `http://www.dem.inpe.br:8000/htbin/sending/scd-ii/Data-Processing-Systems/2/3_dp2.pdf`

2. "European Space Information Technology in the 21st Century, Workshop Proceedings", September 27-28, 2000, European Space Operations Centre (ESOC), Darmstadt, Alemania

3. "Seminar: The Role of Open-Source Software in the Space Business: Technical Issues, Use Guidance, Legal Implications". October 5, 2000, ESTEC, Noordwijk, Holanda