

Integración de SEXTANTE en GearScape.

F. González Cortés⁽¹⁾, V. Olaya Ferrero⁽²⁾

⁽¹⁾ Proyecto GearScape

⁽²⁾ Sextante Geospatial Services S.L. volayaf@gmail.com

RESUMEN

SEXTANTE es un marco para el desarrollo de algoritmos dedicados al procesamiento de información geográficamente referenciada, que actualmente cuenta con más de doscientos algoritmos que son capaces de operar sobre datos vectoriales, alfanuméricos y raster.

Por otra parte, GearScape es un sistema de información geográfico orientado al geoprocesamiento, que dispone de un lenguaje declarativo que permite el desarrollo de geoprocesos sin necesidad de herramientas de desarrollo complejas. Dicho lenguaje está basado en el estándar SQL y extendido mediante la norma OGC para el acceso a fenómenos simples. Al ser un lenguaje mucho más simple que los lenguajes de programación imperativos (java, .net, python, etc.) la creación de geoprocesos es también más simple, más fácil de documentar, menos propensa a bugs y además la ejecución es optimizada de manera automática mediante el uso de índices y otras técnicas.

La posibilidad de describir cadenas de operaciones complejas tiene también valor a modo de documentación: es posible escribir todos los pasos para la resolución de un determinado problema y poder recuperarlo tiempo después, reutilizarlo fácilmente, comunicárselo a otra persona, etc. En definitiva, el lenguaje de geoprocesamiento de GearScape permite "hablar" de geoprocesos.

La integración de SEXTANTE en GearScape tiene un doble objetivo. Por una parte se pretende proporcionar la posibilidad de usar cualquiera de los algoritmos con la interfaz habitual de SEXTANTE. Por la otra, se pretende añadir al lenguaje de geoprocesamiento de GearScape la posibilidad de utilizar algoritmos de SEXTANTE. De esta manera, cualquier problema que se resuelva mediante la utilización de varios de estos algoritmos puede ser descrito con el lenguaje de geoprocesamiento de GearScape.

A las ventajas del lenguaje de GearScape para la definición de geoprocesos, se añade el abanico de geoprocesos disponible en SEXTANTE, por lo que el lenguaje de geoprocesamiento de GearScape nos permite "hablar" utilizando vocabulario de SEXTANTE.

Palabras clave: SEXTANTE, GearScape, geoanálisis, geoprocesos

ABSTRACT

SEXTANTE is a framework for geospatial analysis, currently including a set of more than 200 gegorithms for both raster and vector data, and a set of base classes upon which new algorithms can easily be implemented.

GearScape is a geoprocessing-oriented Geographical Information System which features a declarative language that allows for the creation of geoprocessing algorithms with no need for programming. This language is based on the SQL standard and extended using OGC's Simple Features Standard. Being much simpler than imperative programming languages (i.e. Java, Python, etc), it allows for an easier creation of new gegorithms, these being easier to document, less bug-prone, and highly efficient, since they can make use of built-in optimization mechanisms such as spatial indices.

Integrating SEXTANTE and GearScape is done in two different ways. First, SEXTANTE gegorithms are incorporated into GearScape using the standard SEXTANTE interface, thus enriching the array of different analysis techniques available to GearScape users and providing and almost seamless transition from other GIS for users with previous knowledge of SEXTANTE. Second, the GearScape processing language is extended so it can be used to programmatically execute SEXTANTE algorithms. This way, any problem that can be solved using SEXTANTE gegorithms could be described using the GearScape processing language. This extended language will feature the powerful semantics of GearScape and the rich vocabulary of SEXTANTE, constituting a flexible tool for solving most geo-analysis problems.

Key words: SEXTANTE, GearScape, geoanalysis, geoprocessing

INTRODUCCIÓN. PRESENTACIÓN DE PROYECTOS

GearScape es un sistema de información geográfico (SIG) de escritorio cuyo objetivo no es proporcionar una gran cantidad de funcionalidades al usuario, sino hacer el desarrollo de éstas sencillo para usuarios avanzados y desarrolladores. Una de las líneas seguidas para tratar de cumplir dicho objetivo, ha sido la creación del lenguaje de geoprocésamiento de Gearscape o GGL (Gearscape Geoprocessing Language) que permite la definición de geoprocésos de manera sencilla. Dicho lenguaje está basado en el estándar SQL'92[1] y extendido espacialmente mediante la norma OGC para el acceso a fenómenos simples[2][3]. Al ser un lenguaje mucho más simple que los lenguajes de programación imperativos (java, .net, python, etc.) la creación de geoprocésos es también más simple, más fácil de documentar, menos propensa a bugs y además la ejecución se optimiza de manera automática mediante el uso de índices y otras técnicas.

Para hacer posible la ejecución de instrucciones SQL con fuentes de datos heterogéneas como las que pueden ser usadas en GearScape (shapefiles, bases de datos remotas, etc.), es necesario proporcionar un mecanismo que abstraiga el formato y la localización de las fuentes de datos que procesa[4]. Cuando una fuente de datos se registra en el sistema, ya sea un shapefile local o una tabla en una base de datos remota, se crea un identificador único que sirve para referenciar dicho origen de datos en los scripts GGL, consiguiendo así independencia del formato y la ubicación de los datos usados en ellos. De hecho los resultados de dichos scripts son almacenados por defecto en un formato propio y en un directorio de resultados

predeterminado, pudiendo exportar luego estos resultados a formatos más habituales y en la ubicación deseada.

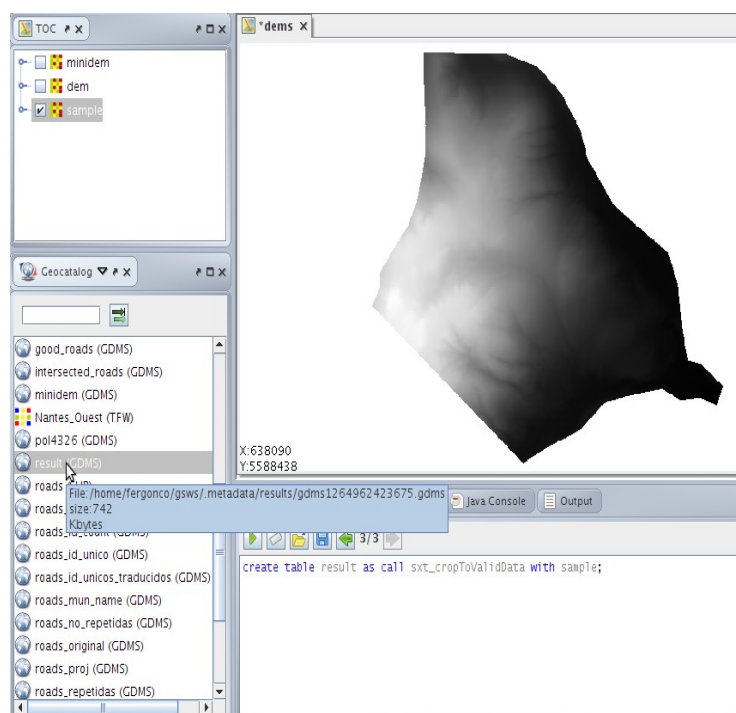


Figura 1: El Geocatalog contiene la lista de identificadores de las fuentes de datos registradas en el sistema.

Además de estar extendido espacialmente, GGL incorpora varias extensiones adicionales realizadas sobre la base SQL que han sido aprovechadas para realizar la integración. La primera es la posibilidad de utilizar tipos de datos raster en las instrucciones GGL, de la misma manera que se tratan otros tipos de datos (strings, enteros, geometrías, etc.). Así pues, un fichero raster se convierte en GGL en una tabla de una fila y una columna cuya única celda contiene el raster.

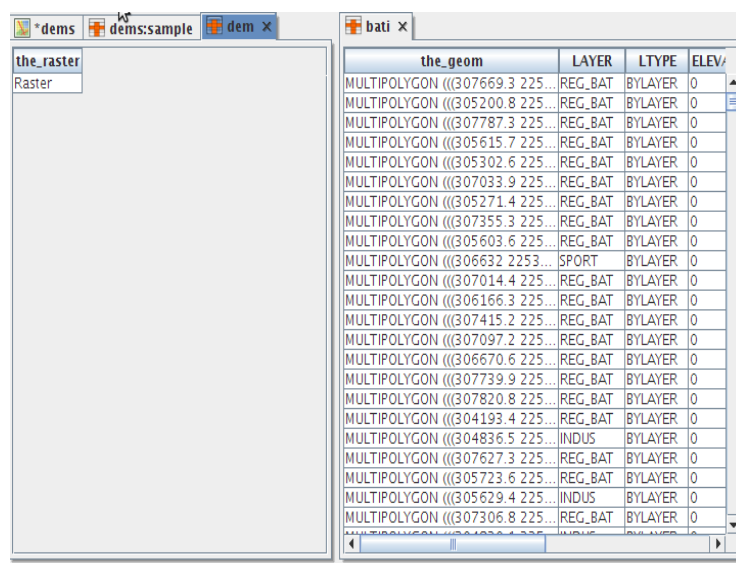


Figura 2: A la izquierda una tabla raster con una fila y una columna. A la derecha una tabla con diversos campos.

El hecho de que los operadores SQL no sean muy útiles a la hora de tratar raster en la manera en la que han sido integrados en GGL, nos ha llevado a la siguiente extensión del lenguaje: las *custom queries*. Una *CustomQuery* no es más que un algoritmo programado en Java que implementa una determinada interfaz y que tienen un nombre por el que se le puede invocar desde GGL. Así, si por ejemplo tenemos un algoritmo llamado *filter* que pone a cero los valores de un raster por debajo de un determinado valor, podríamos invocarlo mediante una instrucción similar a “*CALL filter WITH myraster, 200*”, donde el nombre del algoritmo se especifica tras la palabra clave *call* y los parámetros se ponen separados por comas tras la palabra clave *with*.

Entre las ventajas de la utilización de un lenguaje declarativo como GGL, podemos destacar el hecho de que es posible analizar completamente un script en búsqueda de errores tales como incompatibilidades de tipos, referencias a campos inexistentes, etc. Así pues, en el script de la figura 3 podemos detectar que la segunda instrucción referencia un campo “*my_geom*”, inexistente en la tabla “*temp*”, en lugar del campo “*my_geometry*” que sí que existe. De esta manera se anticipan errores antes de ejecutar y se evitan casos en los que la ejecución falla tras un tiempo largo de ejecución.

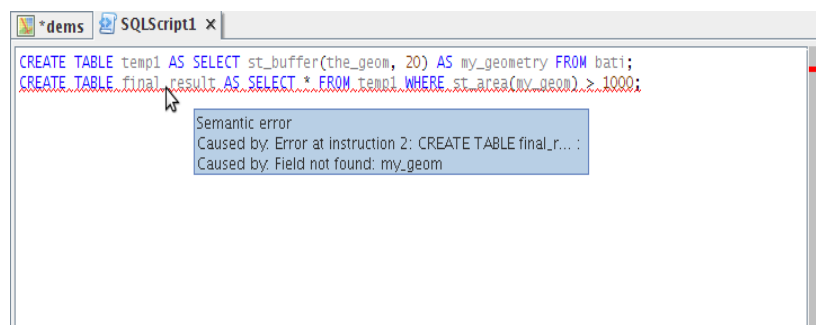


Figura 3: Detección de errores previa a la ejecución.

INTEGRACION ESTÁNDAR DE SEXTANTE

Se ha realizado una integración de SEXTANTE de la manera en la que habitualmente se ha venido haciendo en otros SIG, tales como gvSIG u OpenJump, mediante la inclusión de la ventana de herramientas desde la que el usuario puede seleccionar cualquier algoritmo y ejecutarlo.

Sin embargo, como ya se ha mencionado, GearScape almacena los resultados en un directorio por defecto y les asigna un identificador único. Esto ha provocado que el panel de ejecución de los algoritmos no deje especificar los ficheros en los que se van a almacenar los resultados, sino los identificadores únicos de estos. Al ejecutar, dichos resultados son almacenados en el directorio de resultados predeterminado y añadidos al sistema con el identificador indicado.

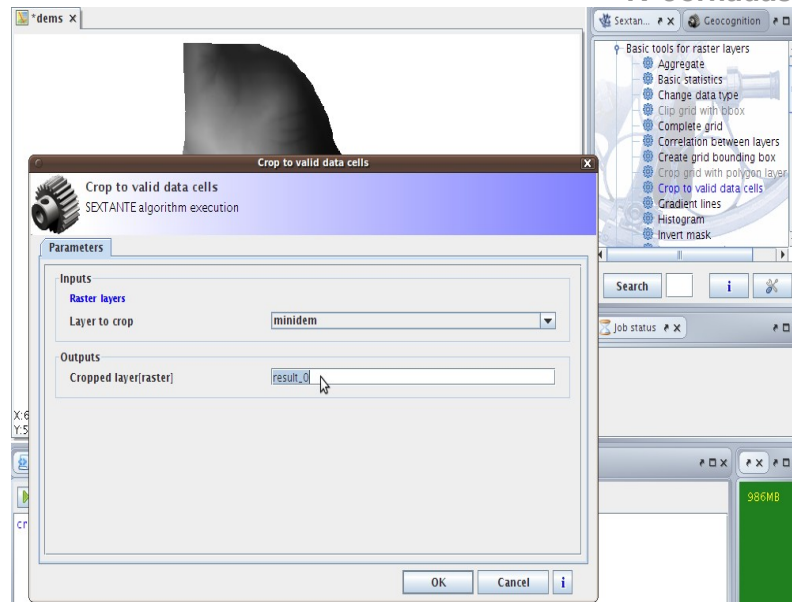


Figura 4: El diálogo de SEXTANTE pregunta por el identificador del resultado en lugar de hacerlo por el fichero donde se van a guardar los resultados.

Otro de los factores que ha influido en la integración ha sido el hecho de que GearScape incorpora una interfaz para la construcción de modelos y un entorno de scripting (GGL). En efecto, no se han incluido en esta integración ni el constructor de modelos de SEXTANTE, ni su entorno de scripting sobre Beanshell.

Además de la integración de la caja de herramientas de SEXTANTE en una vista de GearScape, se han desarrollado los enlaces necesarios entre los objetos de acceso a datos de GearScape y de SEXTANTE que permiten a los *GeoAlgorithm* acceder a las fuentes de datos de GearScape y añadir resultados al sistema.

INTEGRACIÓN DE SEXTANTE EN GGL

Con el fin de integrar SEXTANTE en GGL se ha explotado la funcionalidad para la invocación de algoritmos escritos en Java. Para ello se ha creado una implementación de la interfaz *CustomQuery* que extrae la información de los *geoalgoritmos* de SEXTANTE para satisfacer los requerimientos de dicha interfaz. Con el fin de evitar colisiones entre los nombres de los algoritmos de SEXTANTE y los de las *custom queries* de GearScape se ha optado por añadir el prefijo "stx_" a los nombres de los primeros. De esta manera, un algoritmo de SEXTANTE puede ser invocado de la misma manera que se invoca una *CustomQuery*, utilizando la palabra clave *call* y especificando todos los parámetros tras la palabra clave *with*, como se puede ver en la figura 1.

MODIFICACIONES Y ADAPTACIONES

La integración con un sistema como GearScape ha sido complicada en diversos aspectos. El primero de ellos tiene que ver con el requisito de que los resultados en GearScape deban tener nombres únicos ya que en SEXTANTE las salidas toman un nombre descriptivo que no tiene por qué serlo. Dicha incompatibilidad se ha resuelto mediante la modificación de la clase *GeoAlgorithm* de SEXTANTE para que acepte la especificación del nombre que deberá tomar cada uno

de los resultados. Para conservar la compatibilidad hacia atrás, en caso de que no se especifiquen estos nombres el comportamiento es el habitual.

Sin embargo, los principales problemas, aún por resolver, están relacionados con las diferencias entre las *CustomQuery* de GearScape y los *GeoAlgorithm* de SEXTANTE. La primera de ellas es que mientras que una *CustomQuery* puede devolver sólo un resultado en forma tabular, un *GeoAlgorithm* es capaz de devolver múltiples resultados. Para evitar dicha incompatibilidad, y dado que hay prevista una evolución de GearScape que permitirá a las *custom queries* devolver múltiples estructuras tabulares, se ha optado por excluir de la integración a aquellos algoritmos de SEXTANTE que devuelven más de un resultado.

La segunda y más importante tiene que ver con la capacidad de las *custom queries* de informar, antes de que tenga lugar la ejecución, sobre la estructura de campos de un resultado en base a las estructuras de campos de los parámetros de entrada.

Como hemos visto anteriormente, es posible analizar los scripts GGL previamente a su ejecución para detectar lo más pronto posible errores cometidos por el usuario. Los operadores SQL estándar tienen un comportamiento predefinido por lo que es fácil saber qué estructura de campos va a tener determinada operación SQL. Así, por ejemplo el resultado de la instrucción *"SELECT * FROM mitabla WHERE micampo=1"* tendrá todos los campos de *mitabla*, y el de la instrucción *select st_buffer(the_geom, 20) as geo from mitabla* tendrá un campo llamado *geo* de tipo geométrico. Sin embargo, cuando la instrucción GGL es una llamada a una *CustomQuery* tal que *"CALL mi_query WITH 200, 200"* no tenemos medio posible de saber qué estructura tendrá el resultado, a no ser que el código java que implementa la operación nos aporte la información, como efectivamente sucede.

Dado que los algoritmos de SEXTANTE no proporcionan dicha información, se ha desarrollado un programa que los ejecuta con distintos datos de entrada y analiza las distintas salidas para deducir el comportamiento de los mismos y escribir un fichero XML con las salidas de los algoritmos en función de las entradas. Una de las consecuencias positivas de dicho análisis ha sido la detección de varios bugs existentes en SEXTANTE.

Al contrario que la integración estándar, en la que están disponibles todos los algoritmos, la integración en GGL carece de aquellos algoritmos que no pudieron ser analizados, aproximadamente un 10%, y de ellos, sólo se han podido integrar aquellos que producen un único resultado. En total hay unos 180 algoritmos integrados, de los cuales alguno puede dar problemas al ser incluido en scripts de más de una instrucción debido a una mala determinación de la estructura de sus resultados.

VENTAJAS

Como hemos visto al principio del artículo, GearScape es un entorno con grandes capacidades de análisis gracias a su lenguaje GGL, el cual permite la creación de geoprosos sin necesidad de utilizar entornos de desarrollo ni lenguajes de programación complicados para usuarios sin perfil de programador. Sin embargo, actualmente no dispone de un amplio número de geoprosos disponibles para la realización de análisis.

Por el contrario, SEXTANTE cuenta con una gran cantidad de algoritmos disponibles para realizar análisis pero es necesario tener conocimientos de programación en java para crear algoritmos nuevos.

La integración de SEXTANTE en GearScape dota a este último de un gran abanico de algoritmos con el que procesar sus datos, además de la capacidad propia de generar nuevos, convirtiéndose en una plataforma muy adecuada para el análisis. La integración con el lenguaje GGL permitirá a los usuarios escribir scripts en los que enlazar llamadas a algoritmos de SEXTANTE y/o operaciones de SQL espacial estándar, obteniendo así un lenguaje de geoprosesamiento flexible y muy potente.

CONCLUSIONES Y TRABAJOS FUTUROS

Uno de los trabajos que se pretende realizar consiste en la evolución de GearScape hacia un Sistema de Gestión de Bases de Datos (SGBD) completo. Actualmente, GearScape proporciona una implementación del lenguaje de consulta y de manipulación SQL'92 (SELECT, UPDATE, DELETE, ...) , careciendo de la operaciones y semántica propias del lenguaje de definición de datos tales como los operadores CREATE TABLE, claves ajenas, etc. Además de estas operaciones se espera implementar un sistema de esquemas, en el que se puedan agrupar tablas pertenecientes al mismo modelo de datos, de la misma manera que sucede en los SGBD más comunes, de manera que una *CustomQuery* sea capaz de devolver, no sólo una tabla, si no un modelo de datos completo. En tal entorno, un algoritmo de SEXTANTE que devuelva varios resultados, podrá ser devuelto como esquema conteniendo distintas fuentes de datos.

En este aspecto, debe señalarse que la integración de SEXTANTE en GearScape difiere notablemente de la integración en otros SIG, ya que plantea un contexto distinto con posibilidades y necesidades diferentes. Además de poner de manifiesto algunas carencias en la arquitectura de SEXTANTE para responder a las necesidades de GearScape, ha aportado elementos que abren la puerta a nuevos desarrollos o que esperamos que faciliten futuras tareas que mejoren la estabilidad y capacidades de SEXTANTE. Entre ellas, es de destacar que el análisis sistemático de los algoritmos de SEXTANTE que se ha realizado es de suma importancia, en especial para las tareas de testeo automatizado que se han empezado a incorporar y que, dado el número elevado de algoritmos existentes, suponen un gran esfuerzo si no se recurre a mecanismos automatizados para facilitar una parte de su contenido.

La intención de SEXTANTE es encaminar parte del desarrollo a implementar los métodos y funcionalidades necesarias para que la integración con GearScape sea completa. Esperamos que con ello no sólo se beneficie GearScape en sí, sino también otras aplicaciones con las que ya existan o puedan existir conexiones. El desarrollo más importante que se ha de afrontar es la definición detallada de las salidas de los algoritmos, ya que la semántica actual no es lo suficientemente completa. Una semántica más completa y robusta permitiría a GearScape el uso pleno de SEXTANTE a través de GGL, pero también posibilitaría incorporar modificaciones en el modelizador de SEXTANTE, evitando por ejemplo parte de las comprobaciones en tiempo de ejecución, pudiendo hacerse éstas en tiempo de diseño.

Por su parte, la librería de acceso a datos de GearScape es independiente de éste y ha demostrado un rendimiento excelente, por lo que una línea de desarrollo a considerar es generar nuevas soluciones basadas en SEXTANTE y dicha librería únicamente, de modo similar a la versión de SEXTANTE en línea de comandos que existe en la actualidad basada en GeoTools.

REFERENCIAS

- [1] ISO, Structured Query Language – 92, International Organization for Standardization, 1992.
- [2] OGC, OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 1: Common architecture, Open Geospatial Consortium, 2006.
- [3] OGC, OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option, Open Geospatial Consortium, 2006.
- [4] E. Bocher, T. Leduc, G. Moreau, F.G. Cortés. GDMS: An abstraction layer to enhance SpatialData Infrastructures usability, Proc. of Agile 2008, Girona, Spain 2008.