Extracción de Información a partir de Catálogos Web de Componentes para SIG

Gabriela Gaetan[1], Alejandra Cechich[2], Agustina Buccella[2]

[1]Proyecto de Investigación Área Ingeniería de Software
Unidad Académica Caleta Olivia - Universidad Nacional de la Patagonia Austral
ggaetan@uaco.unpa.edu.ar

[2]Grupo de Investigación en Ingeniería de Software del Comahue (GIISCo)
http://giisco.uncoma.edu.ar
Departamento de Ciencias de la Computación -Universidad Nacional del Comahue
{abuccel, acechich}@uncoma.edu.ar

Resumen. Uno de los problemas en el crecimiento del desarrollo de software basado en componentes es la dificultad para localizar y recuperar componentes de software existentes. Para que los desarrolladores puedan realizar las búsquedas rápidamente e identificar los componentes candidatos, es necesario organizar y catalogar las colecciones de componentes de software en estructuras que posibiliten la facilidad de las búsquedas. Las características particulares de los Sistemas de Información Geográficos (SIG) provocan que las composiciones de componentes no puedan construirse sólo como simples piezas de un producto, sino como un conjunto de elementos pre-ensamblados lo que hace la identificación de candidatos particularmente compleja. En ese contexto, este artículo presenta un sistema para extraer información sobre componentes SIG publicada en repositorios Web, estructurándola en base a un esquema de información estandarizado y enriquecido por medio de técnicas de Procesamiento del Lenguaje Natural. Se describen los principales elementos del proceso y se ilustra con un caso de estudio.

1 Introducción

El Desarrollo de Software Basado en Componentes (DSBC) procura la reducción del tiempo de desarrollo, de los costos y del esfuerzo, y también la mejora de la calidad del producto final debido a la reutilización de componentes software ya desarrollados, probados y validados. La selección de componentes OTS (Off-The-Shelf) se puede ver como un proceso compuesto en el que intervienen por un lado el desarrollador del componente y por otro, el desarrollador de la aplicación. El desarrollador del componente es el responsable de suministrar la información necesaria para la posterior búsqueda, entendimiento y decisión sobre el uso del componente desarrollado.

En la literatura se encuentran distintos aportes relacionados con la clasificación, búsqueda y recuperación de componentes [2]. Algunos trabajos [1,3,7,9,12,13], proponen resolver los problemas relacionados con la categorización (o indexación), concentrando sus propuestas principalmente en las estructuras de clasificación de la información sobre componentes. Otros trabajos, además de realizar propuestas de clasificación para componentes, avanzan en la solución de los problemas de búsqueda y almacenamiento de información sobre los componentes. Por ejemplo, el sitio web Componex[1], permite realizar la búsqueda en su repositorio al organizar y clasificar los componentes de software, usando un esquema XML bien definido para especificar distintos parámetros (tecnología, plataforma, dominio, etc.). ComponentXchange [14] es una infraestructura enfocada tanto en la especificación como en la búsqueda y localización de los componentes que mejor satisfagan los requerimientos de los integradores. Los documentos de especificación de los componentes son enviados por los vendedores y se los almacena en un repositorio de descripción de componentes.

En [4], hemos analizado el aporte de estos trabajos para definir un esquema de clasificación estándar aplicado a un dominio particular; como resultado hemos presentado una ontología de componentes para SIG, que beneficie tanto a desarrolladores como a "reusadores".

Otras propuestas, como en [11], proponen un enfoque basado en ontologías y modelos de dominio, orientado a incrementar la efectividad de la búsqueda y brindar información que relacione a los componentes recuperados. En [6] se describe un sistema para clasificación y recuperación que extrae automáticamente, a partir de las descripciones del software, el conocimiento necesario para catalogar los componentes en la base de conocimiento usando técnicas de procesamiento de lenguaje natural.

Todas las soluciones enumeradas se enfocan sólo en algún problema particular relacionado con la recuperación de información de componentes. Algunas soluciones están dirigidas a los aspectos relacionados con la clasificación; mientras que otras están enfocadas en aspectos de documentación y almacenamiento de la información relacionada con los OTS. En [5] hemos presentado una propuesta para desarrollar un sistema de Publicación de componentes, que brinde soporte a todas las funciones necesarias para construir un repositorio de componentes SIG: clasificación, documentación y almacenamiento. El sistema facilita un proceso de clasificación automática a partir de las descripciones de componentes SIG disponibles en catálogos web aplicando técnicas de Extracción de Información.

En este artículo, describimos el proceso de Extracción de información enriquecido con tecnologías asociadas al Procesamiento de Lenguaje Natural y a la Web Semántica. La idea principal consiste en poblar automáticamente un esquema de clasificación normalizado (de aspectos funcionales y no técnicos) de componentes SIG con la información disponible en portales Web especializados. El resto de este artículo se estructura de la siguiente forma: la Sección 2 detalla los principales elementos de la propuesta. La Sección 3 presenta lecciones aprendidas a partir de casos de estudio que analiza un conjunto de descripciones de componentes disponibles en catálogos web, basados en el Sistema de Extracción de información propuesto. Finalmente, la Sección 4 resume conclusiones y trabajo futuro.

[1] http://www.componex.biz

2. Proceso de Extracción Basado en Conocimiento para SIG

Nuestro Sistema de Extracción de Información de componentes permite analizar descripciones de componentes SIG encontradas en catálogos Web que contienen datos detallados en un esquema de clasificación. En [4] publicamos un esquema organizado en 21 categorías distribuidas en tres grupos. En el grupo *Información general y comercial*, se incluyen las categorías: Nombre, Versión, Sitio web, Empresa / desarrollador, e-mail, Teléfono, Dirección postal, Precio, Traducciones, Artefactos, Requerimientos de software, y Requerimientos de hardware. El grupo *Clasificación* contiene: Tipo, Sistema operativo, Lenguaje de programación, Estado, Licencia y Estándares. Y en el grupo *Funcionalidad*: Servicio geográfico, Funcionalidad implementada, y Vocabulario técnico del dominio. Las categorías de clasificación de este esquema surgieron de la comparación y composición de información disponible en catálogos Web que ofrecieran componentes SIG (ComponentSource[2], FreeGIS[3], Freshmeat[4], ESRI[5]) y de la adaptación de la taxonomía de servicios geográficos del estándar ISO/IEC 19119 [8].

El objetivo del Sistema es que, por medio de este proceso, los fragmentos de texto correspondientes a cada uno de los elementos que caracterizan a los componentes sean descubiertos y documentados automáticamente como instancias del esquema de clasificación definido. Cada instancia es almacenada en un repositorio de reuso.

Este Sistema se basa en una ontología (*compoSIG*) y aplica herramientas de procesamiento de lenguaje natural (NLP) para extracción de información encontrada en catálogos web de componentes del dominio de los SIG. Para la implementación usamos GATE[6] (General Architecture for Text Engineering), una de las herramientas de NLP más maduras y ampliamente usadas. Este framework proporciona un conjunto de módulos reusables para la construcción de aplicaciones basadas en procesamiento de lenguaje natural. Las tareas de extracción de información (tokenización, etiquetado semántico y particionamiento de frases) se implementan con ANNIE (a Nearly-New Information Extraction System)[7], un sistema de extracción de información distribuido con GATE.

En la Figura 1 se describe el Sistema de Extracción de Información de Componentes propuesto, donde los *módulos* que lo componen se representan con cajas cuadradas y la *información* con cajas redondeadas. El módulo de *Configuración* permite definir los términos del dominio de los componentes SIG y las relaciones entre éstos y la ontología. Este proceso requiere como entrada la ontología y produce como resultado el conjunto de parámetros necesarios para realizar la clasificación de los componentes. El módulo de *Clasificación*, toma como entrada páginas web que describen componentes SIG y los parámetros definidos en el módulo anterior, y genera las anotaciones en formato XML sobre la página web original, aplicando técnicas de anotación basada en ontologías. Estas páginas web anotadas se usan para

[2] http://www.componentsource.com/
[3] http://freegis.org/
[4] http://freshmeat.net/
[5] http://www.esri.com/
[6] http://www.gate.ac.uk/
[7] http://gate.ac.uk/sale/tao/index.html#annie

poblar un Repositorio de reuso que contiene meta-datos para describir estructuradamente los componentes; en el módulo de *Estructuración* se completan automáticamente las facetas del Esquema de Clasificación asociadas a la ontología. Antes de almacenar definitivamente la información estructurada con la descripción de un componente en el Repositorio de Reuso, el módulo de *Evaluación* permite que un experto en el dominio verifique la validez de las anotaciones y determine un nivel de calidad en base a la completitud de la clasificación resultante.

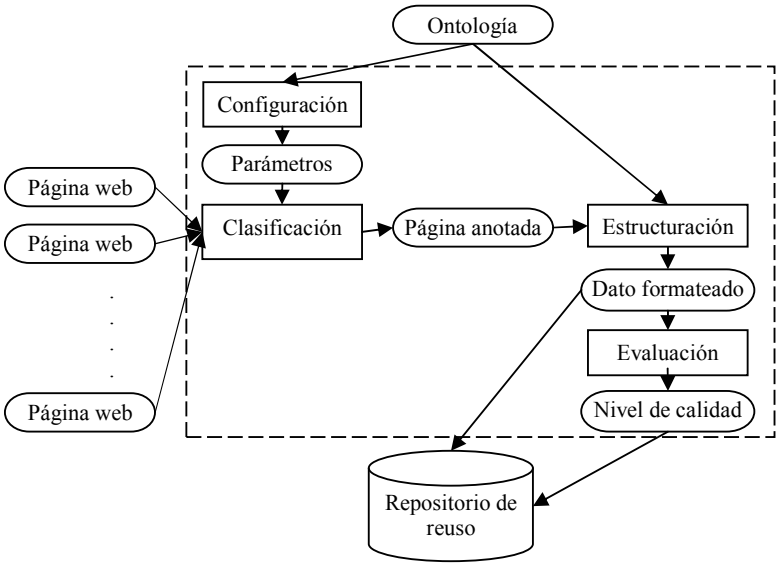


Figura 1. Descripción general del Sistema de Extracción de Información de Componentes.

A continuación se detallan los distintos elementos que forman el Sistema de Extracción de Información de Componentes.

2.1 Ontología

El proceso de Extracción de Información de Componentes requiere como entrada una ontología. En este caso, usamos *compoSIG* [5], una ontología liviana basada en XML que permite la configuración de parámetros para realizar la anotación de las páginas web de componentes SIG.

La Figura 2 muestra las entidades que forman la taxonomía *compoSIG* y que se corresponden con los principales conceptos del dominio de componentes SIG presentados en el Esquema de clasificación propuesto en [4]. Con el objeto de representar información semántica que permita mejorar la recuperación de los componentes, se considera importante incorporar un conjunto de axiomas, entre ellos, las asociaciones generalización/especialización, disyunción, inversa y parte-todo. Aunque las ontologías livianas (pobres en axiomas) presentan algunas deficiencias

por ser menos expresivas, consideramos, al igual que en [10], que una ontología liviana es suficiente para la definición de las entidades, sus atributos y relaciones.

Figura 2. Ontología *CompoSIG*

2.2 Configuración

El módulo de *Configuración*, permite definir los términos más importantes del dominio de los componentes SIG, y sus relaciones con las entidades de la ontología *compoSIG*. Para los propósitos del Sistema de Extracción de Información de Componentes, un experto en el dominio realiza manualmente la configuración de los siguientes parámetros:

1. Las *instancias* de cada concepto de la ontología. Se crea un archivo .lst que contenga las instancias posibles de cada concepto. Por ejemplo, para el concepto 'nombre' creamos un archivo 'nombre.lst', que contiene la lista de los nombres de componentes conocidos.
2. Las *relaciones* entre las *instancias* y los *conceptos de la ontología*. Se genera, en el archivo mapping.def, una asociación entre cada archivo .lst

generado y el concepto de la ontología correspondiente. Por ejemplo, para asociar el archivo 'nombre.lst' al concepto 'nombre' de la ontología *compoSIG* agregamos, en el archivo mapping.def, una línea similar a esta:

```
nombre.lst:file:/C:/Program Files/GATE-5.0-beta1/plugins/ANNIE/resources/gazetteer/compoSIG.xml
```

3. Las *relaciones* entre las *instancias* (definidas en los archivos .lst) y *el tipo de anotación* que se deberá generar. Por ejemplo, para definir el tipo de anotación que corresponde a las instancias del concepto 'nombre', agregamos, en el archivo list.def, una línea similar a esta:

```
nombre.lst:nombre
```

4. El *tipo de anotación*. Como resultado de las definiciones anteriores, todas las instancias sólo pueden ser anotadas con el tipo de anotación 'Lookup'; para poder diferenciarlas, usamos el Jape Transducer[8]. La regla Jape que convierte la anotación 'Lookup' en una anotación de tipo 'nombre', se escribe como:

```
Phase:     nombres
Input: Lookup Token
Options: control = appelt
Rule: nombres({Lookup.majorType == nombre})
:nombre-->:nombre.nombre = {rule = "nombres"}
```

2.3 Clasificación

El módulo de *Clasificación*, permite identificar los términos del dominio, dentro de las páginas web que describen los componentes SIG, por medio de la anotación de entidades nombradas (NE- Named Entities) basada en ontologías.

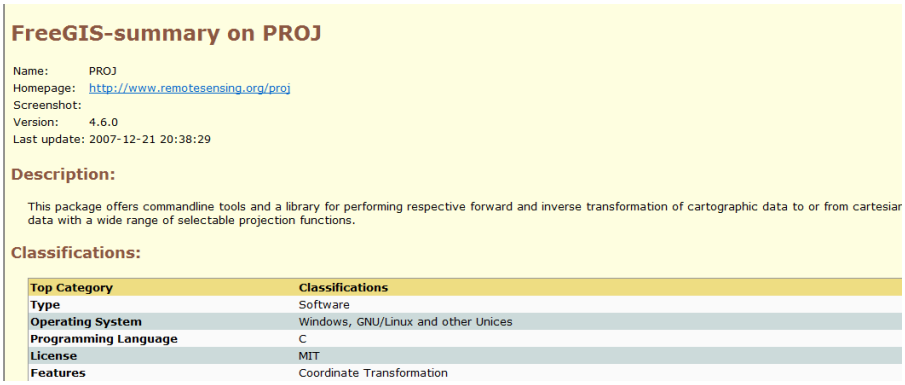
Este módulo está formado por dos sub-módulos:

- *Pre-procesamiento.* En este sub-módulo se identifican los elementos semánticos básicos del texto, y para ello se usan tres componentes de GATE: el Tokenizer, que divide al texto en tokens simples, el Sentence Splitter, que delimita las oraciones del texto y el Part Of Speech Tagger que produce la etiqueta (tag) como una anotación de la categoría gramatical de cada palabra de una oración.
- *Análisis semántico.* Se genera una anotación temporal con un enlace a una clase en la ontología *compoSIG* (por ejemplo, 'Proj' será enlazado a la clase 'nombre' de la ontología). Para cada entidad identificada, se busca la clase que más se adecue en la ontología; si ésta existe, se busca una instancia específica con la cual `concuerde` y se genera el meta-dato que enlaza el contenido de la página web con la ontología. Para determinar si una anotación candidata pertenece a una clase (o sub-clase) de la ontología, se aplican las reglas especificadas con el procesador gramatical (Jape). Este procesador gramatical también permite filtrar anotaciones alternativas (en caso que una referencia de las Entidades Nombradas esté asociada con varias instancias y tipos posibles) aplicando técnicas simples de desambiguación.

[8] Recurso incluido en GATE para manipular anotaciones por medio de gramáticas Jape.

El resultado final del análisis realizado es el contenido de la página web original anotado automáticamente en el formato XML de GATE. En los ejemplos siguientes se muestra el fragmento de una página web en formato original y en formato XML.

Ejemplo1. Formato original de una página web con la descripción del componente PROJ[9].



FreeGIS-summary on PROJ

Name: PROJ
Homepage: http://www.remotesensing.org/proj
Screenshot:
Version: 4.6.0
Last update: 2007-12-21 20:38:29

Description:

This package offers commandline tools and a library for performing respective forward and inverse transformation of cartographic data to or from cartesiar data with a wide range of selectable projection functions.

Classifications:

Top Category	Classifications
Type	Software
Operating System	Windows, GNU/Linux and other Unices
Programming Language	C
License	MIT
Features	Coordinate Transformation

Ejemplo 2. Formato XML de la página web con la descripción del componente PROJ.

```
<GateDocument>
... document's features....
<TextWithNodes>
<Node id="6" />FreeGIS-summary on <Node id="25" />PROJ <Node
id="29" />Name:<Node id="50" />PROJ
... more text ....
</TextWithNodes>
<AnnotationSet>
<Annotation Id="2" Type="Lookup" StartNode="25" EndNode="29">
<Feature><Name className="java.lang.String">ontology</Name>
<Value className="java.lang.String">file:/C:/GATE-5.0-
beta1/compoSIG.xml</Value></Feature>
<Feature><Name className="java.lang.String">class</Name>
<Value className="java.lang.String">nombre</Value></Feature>
<Feature><Name className="java.lang.String">majorType</Name>
<Value className="java.lang.String">nombre</Value></Feature>
</Annotation>
.... more text...
<Annotation Id="6" Type="nombre" StartNode="25" EndNode="29">
<Feature><Name className="java.lang.String">rule</Name>
<Value className="java.lang.String">nombres</Value></Feature>
</Annotation>
.... more text...
</AnnotationSet>
</GateDocument>
```

[9] http://freegis.org/database/viewobj?obj=1092

2.4 Estructuración

El módulo de *Estructuración* usa el texto de las páginas web anotado en el formato XML de GATE para poblar el Repositorio de reuso con información estructurada.

El Repositorio de reuso es una base de datos que no almacena el componente tal como se encuentra en su repositorio original, sino que contiene descripciones basadas en meta-datos de los componentes analizados. Para asegurar el uso de una sintaxis común, el repositorio almacena la descripción del componente basada en los meta-datos en formato XML. La información almacenada permitirá, luego, que el usuario acceda al Repositorio de reuso para encontrar y recuperar aquellos componentes que más se adecuen a sus necesidades. Los meta-datos que describen estructuradamente a cada componente surgen de la ontología *compoSIG*: Nombre, Versión, Sitio web, Precio, Idioma, Artefacto, Requerimientos de software, entre otros. Por razones de espacio, hemos omitido ejemplos de estos meta-datos.

2.5 Evaluación

Hay que notar que antes de almacenar un componente es necesario verificar si los datos completados son consistentes con los datos necesarios en el almacenamiento. Para esto, se verifica la validez de las anotaciones y se hacen las correcciones necesarias. Esta tarea es realizada por expertos en el dominio de los SIG.

Debido a que la información disponible en la web puede no ser suficiente para tomar decisiones arquitectónicas correctas, en este módulo también se propone determinar el nivel de calidad de la información almacenada en el Repositorio de reuso, agregando información sobre la completitud de la clasificación resultante. Esto lleva a incluir procesos de control de calidad que no detallamos en este trabajo.

3. Lecciones aprendidas a partir de casos de estudio experimentales

Con el propósito de validar la capacidad del Sistema de Extracción de componentes para instanciar los conceptos del Esquema de clasificación (representados en la ontología *compoSIG*) a partir de información encontrada en catálogos web, seleccionamos un conjunto de 50 páginas que contienen la descripción de componentes SIG. En este caso de estudio trabajamos con descripciones de componentes SIG publicadas en el sitio FreeGIS[10], donde el texto de las páginas web se encuentra semi-estructurado, está escrito en inglés y, a partir del mismo, es posible definir un vocabulario controlado del dominio.

El resultado obtenido con la versión actual del Sistema de Extracción de componentes (que implementa los módulos de *Configuración*, *Clasificación*, y *Estructuración*) fue un Repositorio de reuso poblado con una instancia del Esquema por cada descripción de componente seleccionada para el caso de estudio.

[10] http://freegis.org/

Tabla 1. Categorías de la ontología *compoSIG* instanciadas con la descripción de un componente.

Categoría de la ontología *compoSIG*	**Datos de la descripción del componente**
nombre	PROJ
version	4.6.0
sitio-web	http://www.remotesensing.org/proj
artefacto	Software
sistema-operativo	Windows, GNU/Linux, other Unices
lenguaje	C
no-comercial	MIT
entrada	cartographic data
salida	cartesian data
tarea	transformation, projection

La instanciación del Esquema de clasificación surge inicialmente relacionando los términos encontrados en la descripción del componente con la correspondiente categoría de la ontología *compoSIG*. Sin embargo, como se observa en el ejemplo de la Tabla 1, no todas las categorías de la ontología se instancian directamente con los datos encontrados en la descripción de un componente (en el caso de la descripción del componente 'Proj' se pudieron instanciar 10 de las 25 categorías). Para enriquecer esta instanciación se agregan reglas gramaticales que capturan patrones de derivación; esto permite definir categorías no requeridas (por ejemplo, la categoría Precio es no requerida cuando Licencia es 'no-comercial') y completar más categorías de la ontología (por ejemplo, la categoría Funcionalidad se puede instanciar con 'processing' cuando la categoría Tarea contiene 'transformation').

4. Conclusiones y Trabajo Futuro

En el presente trabajo hemos detallado y evaluado un proceso de Extracción de información soportado por una herramienta que permite la identificación y recuperación de componentes OTS para SIG a partir de catálogos web usando técnicas del lenguaje natural.

La evaluación de nuestra propuesta se basa en casos de estudio aplicados al análisis de un conjunto de descripciones de componentes encontradas en portales especializados. Actualmente, nuestro trabajo se centra en el desarrollo de una herramienta que nos permita integrar los distintos módulos del Sistema e implementar nuevos casos de estudio. Además, considerando que es posible que la información necesaria para completar el esquema de clasificación no se encuentre en el texto, o que la información se encuentre parcialmente, en el futuro extenderemos las técnicas del proceso presentado a fin de evaluar la calidad de la información almacenada en el Repositorio de reuso.

Agradecimientos. Este trabajo es parcialmente soportado por el proyecto UNComa 04/E072 "Identificación, Evaluación y Uso de Composiciones Software" y el proyecto UNPA 29/B090 "Identificación de Soluciones Off-The-Shelf para Sistemas de Información Geográficos".

Referencias

1. Ackermann, J., Brinkop, F., Conrad, S., Fettke, P., Frick, A., Glistau, E., Jaekel, H., Kotlar, O., Loos, P., Mrech, H., Ortner, E., Overhage, S., Raape, U., Sahm, S., Schmietendorf, A., Teschke, T., Turowski, K.: Standardized Specification of Business Components. German Society of Informatics (2002)
2. Cechich A., Réquilé A., Aguirre J., Luzuriaga J.: Trends on COTS Component Identification. 5th International Conference on COTS-Based Software Systems. IEEE Computer Science Press. Orlando (2006)
3. Dong, J., Alencar, P. S. C., Cowan, D. D.: A Component Specification Template for COTS-based Software Development. First Workshop on Ensuring Successful COTS Development (1999)
4. Gaetán, G., Buccella, A., Cechich A.: Un Esquema de Clasificación Facetado para Publicación de Catálogos de Componentes SIG. XIV Congreso Argentino en Ciencias de la Computación. Chilecito (2008)
5. Gaetán, G., Cechich A., Buccella A.: Aplicación de Técnicas de Procesamiento de Lenguaje Natural y Web Semántica en la Publicación de Componentes para SIG. 38 Jornadas Argentinas de Informática. Mar del Plata (2009)
6. Girardi, M. R.; Ibrahim, B.: A software reuse system based on natural language specifications. 5th International Conference on Computing and Information. Ontario (1993)
7. Iribarne, L., Troya, J. M., Vallecillo, A.: Trading for COTS Components in Open Environments. 27th Euromicro Conference. IEEE Computer Society Press. Warsaw (2001)
8. ISO/IEC. Geographic Information Services. Draft International standard 19119 (2002)
9. Kallio, P., Niemelä, E.: Documented Quality of COTS and OCM Components. 4th ICSE Workshop on Component-Based Software Engineering (2001)
10. Kiryakov A., Popov B., Kirilov A., Manov D., Ognyanoff D., Goranov M.: Semantic Annotation, Indexing, and Retrieval. 2nd International Semantic Web Conference. Florida, USA. (2003)
11. Sugumaran, V.; Storey, V.C.: A semantic-based approach to component retrieval. SIGMIS Database. 34(3): p. 8-24 (2003)
12. Prieto-Díaz, R., Freeman, P.: Classifying Software for Reusability. IEEE Software. 4(1):6-16 (1987)
13. Torchiano, M., Jaccheri, L., Sørensen, C., Wang, I.: COTS Products Characterization. 14th international conference on Software engineering and knowledge engineering. Ischia (2002)
14. Varadarajan, S.; Kumar, A.; Gupta, D.; Jalote, P.: ComponentXchange: An E-Exchange For Software Components. IADIS Conf. WWW/Internet. Portugal (2001)