

Modelado Estructural Espacio-Temporal de Redes de Distribución mediante Componentes de *Software*

William J. Rivas, Jonás A. Montilva
Universidad de los Andes Facultad de Ingeniería
Postgrado en Computación
Mérida, Venezuela
 +58-274-24038111
 {rivasw, jonas}@ula.ve

Resumen— El modelado de redes es una actividad importante para el diseño y construcción de redes de distribución, tales como acueductos, oleoductos y gasductos. Estas redes pueden ser modeladas mediante conexiones de componentes de *software*, los cuales representan elementos reales tales como tanques, válvulas, conductos, etc. Con este fin han surgido en el mercado excelentes herramientas que permiten realizar este tipo de modelado. Algunas de estas herramientas incluyen características de los Sistemas de Información Geográfica (SIG) orientadas a manejar espacial los objetos de la red de distribución.

Uno de los problemas de estas herramientas es que, debido a su carácter comercial, son cerradas; lo cual impide el reuso de sus componentes internos. Por otro lado, la mayoría de ellas funcionan sobre una única plataforma de ejecución, lo que obstaculiza la construcción de aplicaciones distribuidas. El costo de estos artefactos de *software* es elevado y, cuando se requiere su adaptación, el desarrollo de adaptadores requiere de largo tiempo y dominio de la tecnología.

En este trabajo, se describe un conjunto de componentes de *software* reutilizable, que implementan la noción de Grafo Espacio-Temporal, y que pueden ser utilizados para el modelado estructural espacio temporal de una red. La tecnología usada para la implementación de estos componentes es la tecnología CORBA (*Common Object Request Broker Architecture*), la cual es comparada con otras tecnologías, tales como EJB (*Enterprise Java Bean*), COM y DCOM.

Palabras clave— Componentes de *Software*, Reusabilidad, Modelado de Redes de Distribución

I. INTRODUCCIÓN

Una característica de los elementos que forman una red de distribución es que tienen propiedades espaciales y temporales. Las propiedades espaciales permiten ubicar exactamente el elemento en el espacio geográfico. El aspecto temporal se refiere al estudio de propiedades que varían con respecto al tiempo.

El concepto de Grafo Espacial [1],[6] es aplicable en el modelado de redes espaciales. Un Grafo Espacial está formado por elementos que pueden ser nodos y enlaces. Dos nodos espaciales están conectados mediante un enlace espacial. El Grafo Espacial mantiene información topológica definida por la misma naturaleza del grafo lineal plano y soporta operaciones espaciales, tales como ruta mínima, adyacencia de nodos y enlaces y árbol mínimo de expansión. Los elementos de dominio, tales como bombas, válvulas, conductos, tuberías, etc., son representados por el Grafo Espacial en la forma de enlaces. La razón de esta asociación es porque existe una pérdida de energía del fluido transportado causada por la fricción. Un tanque o un embalse se asocian, en principio, a un nodo ya que almacenan energía potencial, aunque, existen nodos en los cuales también existen pérdidas causadas por la fricción.

La caracterización espacio temporal de la red de distribución consiste en modelar dicha red considerando la posición de cada elemento y el tiempo. Además, en el modelado espacio temporal se considera la variación temporal de la forma geométrica del nodo o enlace. Esto permite capturar las variaciones de las formas geométricas de los elementos del dominio, por ejemplo, la forma de un embalse o laguna, forma de conductos de transporte, etc. Para realizar el modelado espacio temporal se aplica el modelo del Grafo Espacio-Temporal [1].

El modelado de las redes, también, incluye la captura de los atributos no-espaciales de los elementos que las conforman. Este tipo de modelado se denomina modelado temático. Cada elemento integrante es considerado como un componente de *software* el cual es comparable a una clase con atributos y operaciones. Evidentemente, el componente posee mayor magnitud en el sentido de que éste puede estar formado por varias clases, además de las interfaces.

El componente de la red es instanciado por el usuario en la base de datos local. Los servicios respectivos de

los componentes son provistos de manera distribuida, es decir, existen instancias de los mismos componentes que proveen servicios de manera transparente a los usuarios. Estos servicios son invocados desde el cliente, de manera remota, por medio de la interfaz del componente.

La configuración anterior ofrece ventajas como la posibilidad de que un grupo desarrollador experto, por ejemplo, en válvulas, ofrezca en un nodo de la red, el despliegue de componentes correspondiente a los diferentes tipos de válvulas con sus funciones y características. Mientras que, en otro nodo de la red del sistema, se despliega las variedades de tanques con sus características.

Las aplicaciones existentes en el mercado que utilizan componentes de *software* para modelar redes de distribución presentan varias restricciones en la interoperabilidad de aplicaciones en sistemas distribuidos. Estas aplicaciones son propietarias, de alto costo, y mayormente, requieren de adaptadores para construir sistemas delegados. En caso de que estas aplicaciones provean componentes de *software*, las mismas implementan la infraestructura para habilitar la operación de componentes. Como ejemplo de estas aplicaciones tenemos Hydrex [4], en la cual se instancian localmente los componentes de manera visual, y *MapObjects* [3], que contiene un conjunto de componentes para el manejo y despliegue espacial. Al considerar las capacidades ofrecidas por *MapObjects* encontramos la dificultad de realizar el modelado temporal, requiriéndose, para esto, el desarrollo de adaptadores para implementar el aspecto temporal. Además, surge la restricción de que sólo pueden funcionar en el sistema operativo Windows.

Los componentes de *software* tienen características fundamentales tales como encapsulamiento, reusabilidad y naturaleza distribuida [7]. La reusabilidad consiste en el empleo reiterativo del componente en el desarrollo de aplicaciones de dominio. La naturaleza distribuida se refiere al uso simultáneo de componentes mediante clientes remotos [7].

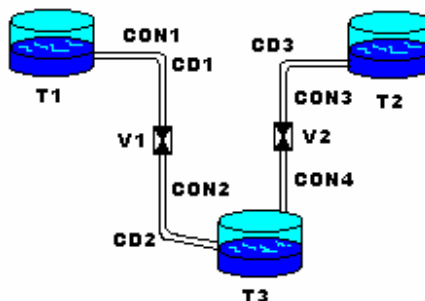
Para permitir la interoperabilidad [2] se considera la tecnología de implementación CORBA, mediante el middleware interno provisto por JAVA 2.

Este artículo está organizado como sigue: en la siguiente sección se describe el concepto de modelado en las redes de distribución. En la sección III se describe el componente RED (Red Espacial de Distribución) para el modelado estructural espacial temporal y temático. En la sección IV se resume las tecnologías consideradas en la implementación. La sección V se presenta algunas de las limitaciones encontradas con la aplicación de las tecnologías de implementación, tales como CORBA y EJB, finalmente, la conclusión y referencias.

II. EL MODELADO EN LAS REDES DE DISTRIBUCIÓN

A. Modelado Estructural

El modelado estructural de una red consiste en definir los diversos elementos que la integran y sus respectivas conexiones con el objetivo de formar su estructura física. Por ejemplo, en la Figura 1 se muestra una red simple (un acueducto) compuesta por varios componentes. Cada extremo de los conductos tiene



ref

conectado otro elemento, el cual pertenece a la categoría nodo.

Figura 1 Modelado estructural de una red

B. Modelado Temporal

En un tiempo inicial $t1$ la red de distribución tiene un aspecto particular debido a la ubicación espacial de cada uno de los elementos que la conforman. Esta distribución espacial puede variar en un tiempo futuro $t2$. La captura de estas variaciones constituye el modelado temporal de la red. Además de esto, en el modelado temporal se captura la variación de la forma geométrica de cada elemento que conforma la red de distribución. Así, es posible el manejo de las variabilidades de la forma geométrica de los elementos, p. ej. en el tiempo $t1$ un elemento puede tener una representación puntual pero en el tiempo $t2$ puede tener una forma poligonal.

De manera similar, como sucede en cartografía, a una escala pequeña de representación existe un nivel mayor de detalle en la forma geométrica que tienen los objetos. Por tanto, las formas de los elementos dependen de la escala de representación.

El Grafo Espacio Temporal permite capturar y describir las variaciones temporales descritas anteriormente. No obstante, con dicho modelo surge el problema de la redundancia de datos, específicamente cuando las propiedades de los elementos no varían. Este problema es resuelto con el componente RED presentado en la sección III.

C. Modelado Temático

El modelado temático consiste en la captura de los

datos temáticos ó no-espaciales de los componentes que el usuario maneja. Por ejemplo, los componentes nodales como los tanques y válvulas tienen la altura como una propiedad temática, los elementos de tipo enlace tales como los conductos tienen como datos temáticos la pendiente, diámetro interno y externo, tipo de material, etc..

Los datos temáticos de los componentes pueden variar o no respecto al tiempo.

III. MODELO DEL COMPONENTE RED PARA EL MODELADO ESTRUCTURAL, TEMPORAL Y TEMÁTICO

Entre algunas de las debilidades presentes en el modelo inicial del Grafo Espacio Temporal tenemos la ausencia de interfaces y la dificultad en el reuso. Para solventar esto se han agregado las interfaces necesarias para permitir su implementación como componente de *software*. El modelo de clases de uno de los componentes se muestra en la figura 2 a cuya implementación se ha denominado componente RED. En la misma figura se pueden observar las relaciones entre las diversas clases que forman el componente.

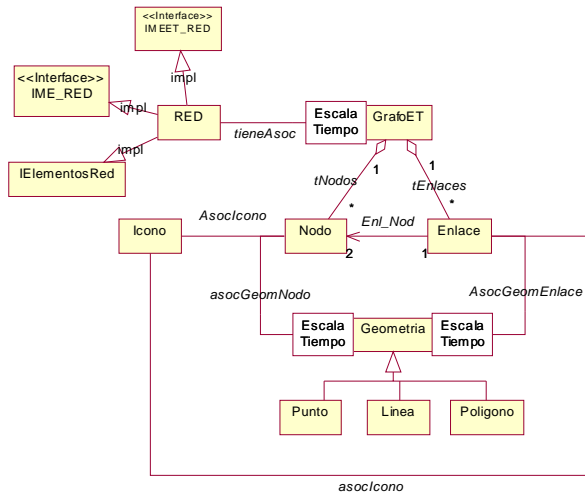


Figura 2 Modelo de clases del componente RED

El componente RED considera los aspectos de modelado estructural espacio temporal de una red de distribución. En el modelado estructural de la red se emplea la interfaz *IME_RED* (Interfaz para Modelado Estructural de Red). En el modelado espacial estructural de un elemento particular se usa la interfaz *IMEE* (Interfaz para el Modelado Espacial Estructural). La interfaz para el modelado estructural de la red (*IME_RED*) se muestra en la figura 3. Las operaciones definidas en esta interfaz permiten realizar inserción y eliminación de componentes a la red, conexiones y desconexiones entre componentes conformantes y la validación estructural. Estas operaciones son implementadas por la clase RED (Figura 2).

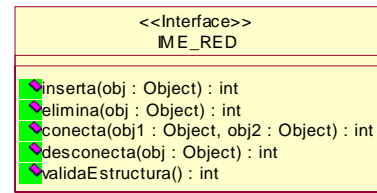


Figura 3 Interfaz para el Modelado Estructural de Red

En el proceso de modelado espacial estructural de los elementos de la red se usa la interfaz *IMEE_* (Interfaz para el Modelado Espacial Estructural). Mediante esta interfaz se define la localización espacial del componente y su aspecto estructural físico. En la figura 4 se muestran las operaciones de esta interfaz para el caso particular de un tanque cilíndrico.

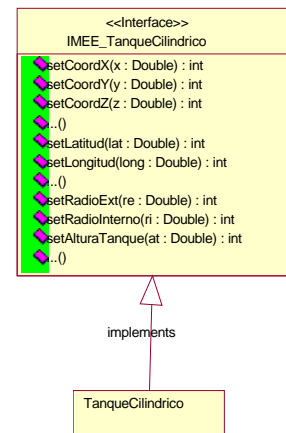
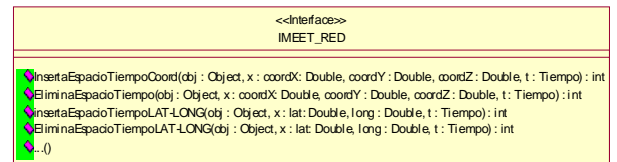


Figura 4 Interfaz para el Modelado Espacial Estructural de un tanque cilíndrico

En el proceso de modelado estructural espacio temporal de la red se emplea la interfaz *IMEET_RED* (Interfaz de Modelado Estructural Espacial Temporal de Red), resumida en la Figura 5. Las operaciones definidas en esta interfaz permiten realizar la captura de la distribución espacial de los elementos a lo largo del



tiempo.

Figura 5 Interfaz de Modelado Estructural Espacial de la Red

IV. TECNOLOGÍAS PARA IMPLEMENTACIÓN DE COMPONENTES

Para la implementación de componentes de *software* han surgido diferentes tecnologías de las cuales entre las más populares destacan COM, DCOM CORBA y EJB.

A. Tecnología COM (Component Object Model)

Un componente COM puede tener varias interfaces por lo cual los archivos de las interfaces coinciden con las definiciones realizadas desde el modelo del componente. En la tecnología COM se requiere implementar obligatoriamente interfaces definidas en la especificación del modelo COM.

Las interfaces construidas son compiladas mediante el un compilador (MIDL) el cual produce archivos de extensión .c, y .h necesarios para la posterior implementación de las interfaces del componente. El servidor se implementa codificando funciones obligatorias (tales como *ClassFactory*, *AddRef*, *Release* y *QueryInterface*) y las funciones lógicas que provee el componente servidor. Para esto se requiere de un conocimiento profundo de la tecnología COM. El cliente es construido utilizando funciones que permitan primero instanciar componentes servidores mediante llamadas de factorías. Igualmente para esto se requiere un conocimiento sólido en el manejo de punteros y procedimientos propios definidos en la especificación COM.

B. Tecnología DCOM (Distributed Component Object Model)

La tecnología DCOM se basa en la tecnología COM. Un componente DCOM usa el protocolo RPC (*Remote Procedure Call*) con el fin de permitir la comunicación entre aplicaciones en diferentes computadoras. Debido a la complejidad en el desarrollo de componentes, se han preparado ambientes con el fin de implementarlos de manera transparente. Por ejemplo, la utilidad llamada MFC (*Microsoft Foundation Class*) provee algunas funciones que permiten construir o implementar las interfaces de componentes de manera automática. No obstante, cuando se requiere realizar una implementación de interfaz de un componente en particular, el desarrollo se dificulta debido a que MFC usa internamente otras APIs que también son complejas.

C. Tecnología EJB (Enterprise Java Bean)

Con el uso de los dos tipos básicos de componentes provistos por EJB (Sesión y de Entidad) es posible construir la mayoría de aplicaciones para el modelado estructural temporal de las redes. No obstante, si se extiende el uso con las especificaciones actuales [9][10], se presenta la posibilidad de lograr un desarrollo difícil de mantener. Esto es debido a la gran cantidad de archivos que deben implementarse tales como archivos de interfaces, de implementación y archivos de configuración.

Cada *bean* de entidad puede compararse como una fila de una tabla, es necesario implementar un *bean* de sesión para que actúe como mediador entre el cliente y el servidor (Figura 6).

Con el fin de evitar la codificación de funciones de administración y proveer mayor simplicidad, cada vez las especificaciones EJB se van actualizando, en donde se agregan características de administración de *beans* en el lado contenedor. Estas características al ser implementadas por algunos propietarios resulta en un *software* particular para el cual se requieren de licencias de alto costo.

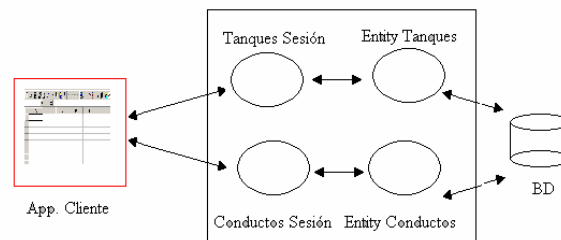


Figura 6 Tipos de *beans* requeridos en el uso de EJB

D. Tecnología CORBA

CORBA es una arquitectura de objetos distribuida propuesta por la OMG [7]. CORBA implementa un modelo Cliente/Servidor donde cada aplicación puede exportar servicios o funcionalidades usando la forma de un objeto. La noción del modelo Cliente/Servidor significa que existe un objeto cliente el cual solicita servicios a otro objeto servidor que provee los servicios requeridos. Con el uso de esta tecnología es posible lograr una interoperabilidad tal como la define Chiang [2]. En la Figura 7 se muestra un diagrama de despliegue con el uso de la tecnología CORBA donde los puertos de escucha son definidos por mismo desarrollador del sistema.

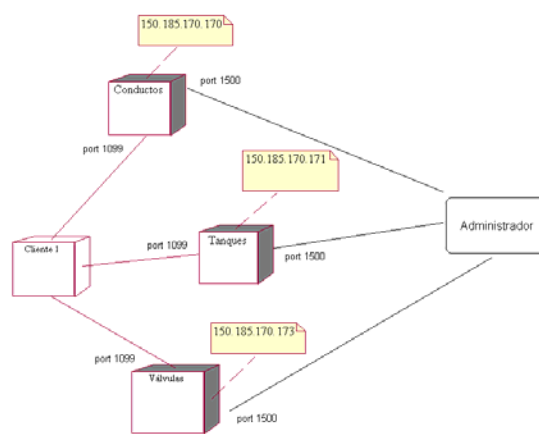


Figura 7 Diagrama de despliegue de componentes CORBA

V. LIMITACIONES ENCONTRADAS EN LAS TECNOLOGÍAS USADAS EN LA IMPLEMENTACIÓN DE COMPONENTES DE SOFTWARE

A. Limitaciones con el uso de EJB

Usando la especificación 2 de EJB [10], para cada *bean* de entidad se requiere de al menos tres interfaces para los *beans* de sesión y al menos 4 interfaces para los *beans* entidad. Esto se convierte en una desventaja al desarrollar sistemas grandes, difíciles de mantener [5]. Además se requiere definir diversos descriptores los cuales algunos dependen del propietario del contenedor. En la tecnología EJB se requiere del buen conocimiento de las características de los contenedores empleados con el fin de evitar mayores errores causados al implementar métodos ya existentes o implementados por los mismos contenedores.

B. Limitaciones con el uso de CORBA

Básicamente, una de las limitaciones en el uso de la tecnología CORBA es la restricción introducida mediante el empleo del lenguaje de definición de interfaces (IDL). Estos lenguajes pueden introducir elementos de software que impiden el uso de un ORB (*Object Request Broker*) abierto. Los tipos de datos usados por estos lenguajes de definición son muy restringidos. En general las operaciones en las interfaces no permiten la aplicación de las técnicas orientada a objetos, p. ej. las operaciones no pueden ser sobrecargadas y el pase de parámetros no puede ser por referencias de objetos. Además, la distribución y la relación entre objetos no se pueden realizar usando el mismo lenguaje.

VI. CONCLUSIÓN

A pesar de las limitaciones citadas anteriormente respecto al uso de la tecnología CORBA, se ha trabajado en base a ella de una manera sencilla para el modelado espacio temporal de redes de distribución. Se tiene una simplicidad del sistema, caracterizado por componentes de *software* que implica una facilidad en el mantenimiento. Los componentes pueden desarrollarse en varios lenguajes de programación para el cual exista un compilador de definición de interfaces compatible con CORBA. Con el lenguaje de desarrollo Java se logra que los componentes operen en diferentes sistemas operativos.

VII. AGRADECIMIENTOS

Se agradece al CONICIT por el financiamiento del presente trabajo registrado bajo el número G-97000824 - proyecto Centauro.

VII. REFERENCIAS

- [1] Besembel I, Montilva J., "Modelling spatio-temporal relationships in object-oriented applications", Proceedings of

- the IASTED International Conference Artificial Intelligence and Soft Computing, 1998, Mexico.
- [2] C. Chiang, "The Use of Adapters to Support Interoperability of Componentes for Reusability", Information and Software Technology, Vol 45, no. 3, March 2003.
- [3] ESRI (2003, September 20). "MapObjects Components" [Online]. Available: <http://www.esri.com/software/mapobjects/index.html>
- [4] Exitech {September 2003}, HYDREX Hydraulic Flow Network Model Builder, [Online] Available: http://www.exitech.com/Models/fluid_hydrex.htm
- [5] Gopalan Suresh Raj. A detailed comparison of CORBA, DCOM and Java [Online] Available: <http://gsraj.tripod.com/>, February 2003.
- [6] Montilva Jonás, "A Spatial model for managing service networks in geographical information systems". Revista Técnica de la Facultad de Ingeniería, Universidad del Zulia, Vol. 23 Número 2, Agosto 2000.
- [7] Object Management Group, Common Object Services Specification, June 2003, <http://www.omg.org>.
- [8] Rogerson D., Inside COM, Microsoft Press, 1997.
- [9] Sun Microsystems, Inc., "Enterprise Java Beans (TM) Specification Version 2.1", 2002.
- [10] Sun Microsystems Inc., "Enterprise JavaBeans Specifications", June 2003, [Online] Available: <http://www.javasoft.com>