

# **Arquitectura de software para Sistemas de Información Ambiental**

**Urciuolo Adriana, Iturraspe Rodolfo, Parson Ariel, Esteban Natalia**

Universidad Nacional de la Patagonia San Juan Bosco – Sede Ushuaia, Darwin y Canga, (9410)

Ushuaia. TE/FAX: 430892. e-mail:

[urciuolo@tdfuego.com](mailto:urciuolo@tdfuego.com), [iturraspe@tdfuego.com](mailto:iturraspe@tdfuego.com), [a-parson@impsat1.com.ar](mailto:a-parson@impsat1.com.ar), [nates@ciudad.com.ar](mailto:nates@ciudad.com.ar)

## **Resumen**

Una característica fundamental de los sistemas de información ambiental es la necesidad de integrar información proveniente de diferentes dominios de conocimiento, mostrando las interacciones presentes entre los diversos componentes de la naturaleza y facilitando el manejo de la complejidad característica de este tipo de sistemas.

El objetivo general del Proyecto de Investigación, es el estudio de técnicas apropiadas para el desarrollo de sistemas de información ambiental, que permitan: manejar la complejidad, representar en forma adecuada la información y simular los procesos relacionados con el ambiente natural, en particular aquellos relacionados con el recurso agua.

En forma específica durante esta etapa del proyecto, se pretende definir una arquitectura de software, que permita el intercambio de información entre distintos sistemas relacionados con el medio ambiente en el marco de una arquitectura SIG y la utilización de la información residente en los mismos por modelos que simulan el comportamiento de la naturaleza.

## **Introducción**

En la actualidad, las organizaciones responsables del manejo ambiental reconocen que la información constituye la base para la toma de decisiones. Se plantea en consecuencia la necesidad de desarrollar sistemas confiables y eficientes, adecuadas para los requerimientos específicos de los mismos (Günther, 98).

La complejidad en el manejo de la información y el modelado de ambientes naturales está presente por muchas razones, entre las cuales se citan (Purbis et al, 99; Günther, 98):

- Gran cantidad de datos a procesar debido al avance continuo de los sistemas automáticos de adquisición de datos.
- Requerimiento del estudio de complejas conexiones lógicas entre datos, dada la diversidad de áreas de estudio interrelacionadas.
- Necesidad de representación espacial y temporal de la información.
- Manejo de datos distribuidos: existe gran cantidad y diversidad de organismos interactuantes que capturan y procesan esta información.
- Los objetos ambientales presentan estructuras complejas
- Necesidad de considerar la influencia humana sobre los ecosistemas.

Teniendo en cuenta estas características, se destacan las siguientes cuestiones a resolver

- Los problemas ambientales pertenecen a un número de diferentes dominios de conocimiento. Son manejados por diferentes sistemas autónomos usualmente heterogéneos tanto técnica como semánticamente (diferentes modelos de datos). A los fines de ser utilizados por los tomadores de decisión en cuestiones ambientales, los datos y la funcionalidad provenientes de diferentes fuentes deben ser preparados para una presentación uniforme (Koschel et al, 96).
- Modelos de simulación utilizan los datos residentes en dichos sistemas, de acuerdo a diferentes estrategias y escenarios de simulación.

En un trabajo fundamental sobre arquitectura de software, Garlan y Shaw (Garlan et al, 94) plantean que a medida que el tamaño y complejidad del software crece, diseñar y especificar la estructura

global de la arquitectura del sistema emerge como un tipo de problema fundamental. Se incluyen cuestiones como organización y control global de la estructura, protocolos para comunicación, sincronización y acceso de datos, distribución física, composición de elementos de diseño, etc.

Larry Bass (Bass *et al.*, 1998), brinda la siguiente definición: “La arquitectura de software de un programa o sistema de computación es la estructura o estructuras del sistema, que comprende los componentes de software, las propiedades externamente visibles de esos componentes y las relaciones entre ellos”.

Obtener la arquitectura adecuada es crucial para el éxito de un sistema de software (Appleton, 00); hoy se reconoce como esencial el contar con una representación arquitectural del sistema, para el análisis y la descripción de las propiedades de alto nivel. Por otra parte, las descripciones arquitecturales han sido reconocidas como esenciales para un sistema bien diseñado (Bass et al, 99).

Existen diversos estilos de arquitectura ampliamente difundidos (pipes and filters, layered, repositorios, etc.) y otros específicos para dominios particulares. Las arquitecturas para dominios específicos de software proveen una estructura organizacional hecha a medida para una familia de aplicaciones. El conocimiento de diseño en un dominio permite definir un estilo arquitectural para una colección de sistemas relacionados, lo cual incluye un vocabulario apropiado para los elementos de diseño del sistema y reglas para su composición. La especialización de la arquitectura al dominio, permite además, simplificar el proceso de construcción de nuevos sistemas, a través de la reutilización de la infraestructura existente, reduciendo costos y facilitando la mantenibilidad de los sistemas (Garlan et al, 95).

Por lo expuesto se considera necesario investigar arquitecturas convenientes para los SIA, que faciliten la construcción de sistemas del dominio, permitiendo la integración entre los mismos, una conveniente representación de la información espacio-temporal, el análisis de los datos mediante tratamientos primarios y estadísticos y la simulación de diferentes fenómenos ambientales. Dado que el Proyecto de Investigación focaliza su atención en sistemas de Hidroinformática, el estudio se realiza fundamentalmente considerando aquellos sistemas ambientales que interactúan con los de información hídrica.

### **Temas de investigación y desarrollo**

Durante esta etapa de la investigación se tomarán decisiones significativas acerca de (Booch et al, 1999):

- La organización del sistema de software
- Los elementos estructurales y sus interfaces, que comprenderán un sistema de información ambiental, junto con su comportamiento tal como se especifica en las colaboraciones entre dichos componentes.
- La composición de los elementos estructurales y de comportamiento en subsistemas progresivamente mayores.
- El estilo arquitectural que guía esa organización: los elementos y sus interfaces, sus colaboraciones y su composición.

Concentrándose en las abstracciones arquitecturales específicas de un dominio, se pueden combinar los mejores aspectos de plataformas estándar y componentes estandarizados para crear y/o especializar sistemas relacionados al dominio. (Garlan et al, 95)

## Actividades del Proyecto

Las etapas iniciales para la definición de la arquitectura se realizan de acuerdo a la metodología propuesta por RUP (Booch et al, 1999) la cual se caracteriza por proponer un proceso de desarrollo de software conducido por casos de uso, centrado en la arquitectura, iterativo e incremental. Según esta metodología, durante la *fase de elaboración* la arquitectura se desarrolla en iteraciones, razón por la cual se considera que puede comenzarse por plantear una arquitectura conceptual básica, que se refine en sucesivos pasos incrementales.

### I. Creación del Modelo del Negocio

Se estudiaron las características y fundamentos de los Sistemas de Información Ambiental (Günther, 98), analizando diferentes estándares utilizados en dominios de aplicación de interés al presente proyecto: Recursos Hídricos, Suelos, Ecología Terrestre, Clima, SIG, Modelación hidrológica y Ambiental.

De acuerdo a RUP, durante la etapa de análisis del dominio es posible construir un Modelo del Negocio, en base a la definición del Modelo de Casos de Uso del Negocio y el Modelo de Objetos del Negocio. Se construyó un Modelo para el Negocio, partiendo de un Modelo del Dominio Físico que muestra las interacciones entre los distintos componentes de la naturaleza, relacionados con los sistemas hidrológicos reales (Urciuolo et al, 03; Urciuolo et al, 02).

### II. Definición de los requerimientos de un SIA

El análisis de los requerimientos se modela en base a casos de uso centrales para este tipo de sistemas. Se definieron como centrales para el estudio de la arquitectura inicial, los casos de uso: “Provisión de Información Ambiental”, “Tratamiento primario de información ambiental” y “Simulación ambiental”, definiendo actores y el Modelo resultante de Casos de Uso.

### III. Estudio de estilos y patrones arquitecturales.

Durante esta etapa se procedió al estudio diferentes estilos arquitecturales (Bass et al, 98). Se estudiaron además diferentes patrones arquitecturales (Buschmann et al, 96; Ericsson, 2000) a los fines de evaluar las ventajas de su aplicación a este tipo de sistemas.

### IV. Estudio de plataformas de integración

Se analizaron plataformas de integración como CORBA de OMG (OMG, 03) y DCOM, a los fines de plantear un Middleware que facilite la resolución del problema de integración y la presentación de datos provenientes de distintos dominios.

### V. Análisis de diferentes arquitecturas existentes para el dominio.

Se analizaron diferentes arquitecturas propuestas para el dominio (Purbis et al., 99; Koschel et al, 96) y en particular, arquitecturas existentes para Sistemas de Información Geográfica (SIG) (Günter, 98), teniendo en cuenta que la integración de cualquier sistema de información y/o de modelación a SIG, constituye actualmente un requisito básico, dadas las facilidades en el ingreso de datos, análisis y presentación de los mismos.

### VI. Creación y/o selección de una arquitectura para el dominio basada en los requerimientos expuestos.

Esta etapa del proyecto se encuentra en desarrollo.

Se estudia en primer lugar la posibilidad de aplicar estilos y patrones arquitecturales existentes. En particular se analizaron los siguientes casos:

### 1) Utilización de un estilo arquitectural Layers:

Partiendo del Modelo del Negocio y de los requerimientos, se consideró conveniente trabajar en niveles de diferente abstracción; por lo tanto se considera adecuada la utilización de un estilo Layers. También conocido como patrón arquitectural (Buschmann et al, 1995) permite estructurar aplicaciones que pueden descomponerse en grupos de subtarefas, trabajando cada uno de ellos en un nivel particular de abstracción. El patrón Layers define cómo organizar el modelo de diseño en capas (Booch, et al, 1999); se presentan subsistemas de aplicación individuales en la capa superior, contruidos a partir de subsistemas en las capas inferiores, tales como frameworks y librerías de clases.

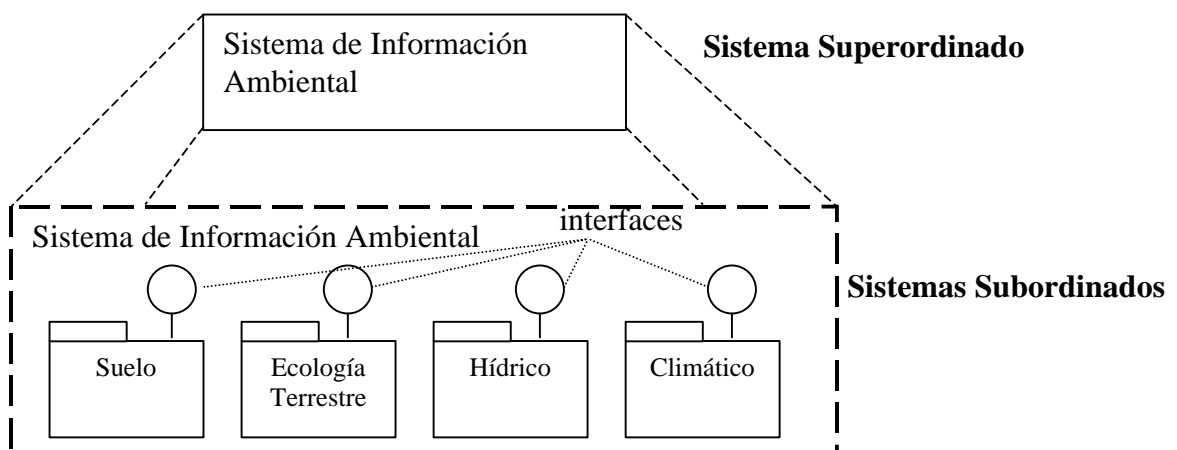
La arquitectura de las capas superiores se crea a partir de los casos de uso arquitecturalmente relevantes y las inferiores como Middleware y System layers contiene capas que no dependen de los casos de uso del Negocio. La capa de aplicación general contiene subsistemas que no son específicos de una aplicación única, pero que pueden ser reutilizados para muchas aplicaciones diferentes dentro del mismo dominio o negocio. En esta capa se ubican los sistemas correspondientes al *Nivel de Información Ambiental*. La capa inmediatamente inferior contiene una arquitectura para SIG, que permite la representación geográfica de las capas superiores. En la capa superior de aplicaciones específicas se ubican los sistemas correspondientes a modelos de simulación (Urciuolo et al, 03). La arquitectura de los dos niveles inferiores: puede establecerse sin considerar los casos de uso, ya que estas capas no son específicas del dominio. En la capa Middleware se ubica la arquitectura correspondiente a una plataforma de integración como CORBA, etc.

### ***Aplicación Patrón arquitectural “Sistema de sistemas interconectados”:***

Uno de los requerimientos planteados fue la necesidad de facilitar la interacción entre los diversos componentes de un SIA. Se utiliza para ello en la capa correspondiente al Nivel de Información ambiental, el patrón arquitectural “Sistema de sistemas interconectados” (Ericsson, 00). Esta construcción es útil cuando se construyen sistemas complejos o que presentan necesidad de integración.

Este tipo de “super sistema” se implementa por un conjunto de sistemas desarrollados en forma independiente, interconectados comunicándose para alcanzar un propósito común. El sistema que representa la capacidad global, se llama “superordinado”, los otros, partes del todo, se llaman “subordinados”. La separación de sistema superordinado de los subordinados tiene claras ventajas:

- ✓ Los sistemas subordinados pueden ser manejados en forma separada durante todas las actividades del ciclo de vida, lo cual es una característica básica del dominio de interés.
- ✓ No es necesaria la construcción del sistema completo. Puede comenzarse por alguno de los subordinados y en otras etapas del ciclo de vida, continuar por los siguientes.



Cada sistema subordinado se desarrolla en la forma usual como una caja negra, considerando los otros sistemas con los cuales se comunica, como actores. Las interfaces a los sistemas subordinados serán propiedad del sistema superordinado de Información ambiental.

En la actualidad se están analizando otros estilos y patrones arquitecturales, a los fines de plantear una arquitectura de software general para el dominio.

## Conclusiones y trabajos futuros

Se ha avanzado en el estudio de una arquitectura de software apropiada para el dominio de interés, en base al estudio de estilos, patrones, arquitecturas y estándares existentes, partiendo de un modelo del negocio que considera las interacciones existentes en el dominio físico.

La utilización de un estilo arquitectural layers permite resolver la cuestión de las interacciones entre sistemas “pares” en una capa interconectados entre sí, permitiendo que en una capa específica de simulación diversos modelos ambientales utilicen la información existente en la capa inferior. En una capa Middleware se ubica la arquitectura correspondiente a plataformas de integración.

Si bien ya han sido analizadas algunas posibles arquitecturas para los requerimientos definidos, se deberá continuar estudiando otros estilos y patrones para obtener una arquitectura general. Una vez evaluadas diferentes alternativas de arquitectura se implementará un sistema basado en dicha arquitectura, a los fines de obtener una evaluación ajustada.

## Bibliografía

- ◆ Appleton B. *Patterns and Software: Essential Concepts and Terminology*. Página Web: [www.enteract.com/~bradapp/](http://www.enteract.com/~bradapp/), 2000
- ◆ Bass L., Clemens P., Kazman R. *Software Architecture in Practice*, 1998
- ◆ Bass L., Kazman R. *Architecture-Based Development*. Technical Report CMU-SEI-99-TR-007, 1999
- ◆ Booch G., Jacobson I., Rumbaugh J. *The Unified Process Software Development*. Addison-Wesley Publications, 1999
- ◆ Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P. and Stal, M. *Pattern-Oriented Software Architecture: A system of patterns*. New York: John Wiley & Sons, 1996.
- ◆ Ericsson M, *Developing Large-Scale Systems with the Rational Unified Process*. Rational Software White Paper, 2000
- ◆ Garlan D., Perry *Introduction to especial issue in software architecture*, 1995
- ◆ Garlan D., Shaw Mary. *An Introduction to Software Architecture. Advances in Software Engineering and Knowledge Engineering*, Volume I, World Scientific Publishing Company, New Jersey, 1994.
- ◆ Günther O. *Environmental Information Systems*. Springer-Verlag, Berlín, Germany, 1998
- ◆ Koschel A., Kramer, R. *A Federation Architecture for an Environmental Information System incorporating GIS, WWW and CORBA*, Universität Karlsruhe, Germany, 1996
- ◆ OMG (Object Management Group) Web Page. CORBA BASICS, 1997-2003.
- ◆ Purbis M., Cranefield S. *A Distributed Architecture for Environmental Information Systems*, New Zeland, 1999
- ◆ Urciuolo A., Iturraspe R., Sandoval S., Parson A. *Estudio de Técnicas apropiadas para modelar sistemas de Hidroinformática en el contexto de los Sistemas de Información Ambiental*, WICC 2002, P 339-343.
- ◆ Urciuolo A., Iturraspe R. *Conceptual Patterns for Water Information Systems*, Journal of Computer Science & Technology, 2003.