

Agentes y Bases de Datos Activas: Un modelo formal de propósito general y una aplicación al registro y consulta de datos y metadatos científicos ambientales

Propuesta Doctoral

Virginia Padilla Sifontes

Estudiante Doctoral del Centro de Investigación y Proyectos en Simulación y Modelos

Universidad de Los Andes, Mérida, Venezuela.

vpadilla@uneg.edu.ve

Tutor:

Dr. Jacinto Dávila

jacinto@ula.ve

Centro de Investigación y Proyectos en Simulación y Modelos

Universidad de Los Andes, Mérida, Venezuela.

Resumen

Este es un proyecto de investigación en modelado de bases de datos y conocimiento donde nos proponemos desarrollar un modelo formal basado en Agentes que describa las facilidades de un Sistema de Base de Datos Activa de propósito general. El modelo debe manejar datos complejos, específicamente datos y metadatos sobre sistemas ambientales o ecosistemas y debe servir como soporte para el modelaje de la Biocomplejidad. Con el modelo formal esperamos también poder estudiar la relación entre las Bases de Datos Activa y los sistemas basados en ontologías que caracterizan a la Web Semántica.

Este trabajo está enmarcado en el Proyecto que llevan adelante Venezuela y Estados Unidos de América, para el Modelaje de la Biocomplejidad. Las Universidades participantes son la Universidad de Los Andes (Mérida, Venezuela) y Universidad Nacional Experimental de Guayana (Ciudad Guayana, Venezuela) y la Universidad de North Texas (Texas, EEUU) bajo el auspicio de NSF's Internacional Program (National Science Foundation) y el Fonacit.

La fecha estimada de culminación de este trabajo de investigación es el año 2009

Palabras Claves: Base datos activas, Agentes, Ontologías, metadatos científicos

Introducción. Nos proponemos desarrollar un modelo formal basado en Agentes que describa las facilidades de un Sistema de Base de Datos Activa de propósito general. Como modelo formal, queremos referirnos a una descripción lógica matemática de una base de datos con agentes caracterizando la dinámica que la convierte en una base de datos activa. Un modelo así podría servir, no solamente para describir y comparar sistemas manejadores de bases de datos activas ya existentes, sino para especificar

futuros sistemas, incluyendo algunos que aprovechen la tecnología de agentes para prestar servicios inéditos como los que requieren grandes almacenes de conocimiento, los simuladores de sistemas complejos y los sistemas basados en ontologías. De hecho, nos proponemos evaluar el modelo implementando prototipos funcionales de almacenes de datos y metadatos de relativa complejidad. Mas aún, esos prototipos funcionales implementarían variantes regulares del modelo general que pueden servir como mecanismos de evaluación (*benchmarking*) de diversos sistemas manejadores de bases de datos.

Por esta última razón, este proyecto se inscribe en el dominio de los estudios de Biocomplejidad, como área de aplicación. El modelo debe facilitar la caracterización de datos complejos, específicamente datos y metadatos ambientales y debe servir como soporte para el modelaje de la Biocomplejidad¹. La biocomplejidad ambiental [33] se refiere a la interacción compleja entre los sistemas vivientes y su ambiente, reflejando las propiedades que surgen de esta interacción, haciendo énfasis en la riqueza de sistemas ecológicos y su capacidad por la adaptación y la auto-organización. Es un dominio de conocimiento que plantea claros desafíos para las bases de datos en razón de los volúmenes típicos de datos que se deben manipular, pero también por la variedad de relaciones, transformaciones y consultas que experimentan esos datos en su cultivo y uso regular en la investigación científica.

A Continuación, precisamos la definición de Base de Datos Activa a partir de las referencias más conocidas en la literatura y, seguidamente, realizamos una breve introducción a la llamada teoría de Agentes de la Inteligencia Artificial. Bases de Datos Activa y Agentes Inteligentes se convierten así en los conceptos fundamentales en este proyecto.

Bases de Datos Activas. En [14] se define un Sistema de Base de Datos Activa como uno que proporciona funcionalidad adicional para especificar reglas activas. Esquemáticamente, un Sistema de Base de Datos Activa:

- Es un Sistema de Base de Datos
- Proporciona la capacidad de *monitorear* y *reaccionar* a circunstancias específicas que son relevantes a una aplicación.

El primer punto nos indica que el Sistema de Base de Datos Activa posee un Sistema Manejador de Base de Datos y la propia Base de Datos. Un *Sistema Manejador de Bases de Datos* es un sistema de software cuyo propósito es definir, construir, mantener y operar, de manera confiable y eficiente, bases de datos multi-usuarios

El segundo punto se refiere a la reactividad del sistema. La reactividad como una propiedad de sistemas puede ser enfocada, según [12], desde dos aspectos:

- Reactividad de la Arquitectura: capacidad del sistema de procesar las entradas en tiempo real y dar una respuesta oportuna. Procesamiento de entrada en tiempo real significa que el sistema debe

¹ Este proyecto está enmarcado en el Proyecto Nacional de Modelaje de la Biocomplejidad que llevan adelante la Universidad de Los Andes (Mérida, Venezuela) y Universidad Nacional Experimental de Guayana (Ciudad Guayana, Venezuela) y la Universidad de North Texas (Texas, EEUU) bajo el auspicio de NSF's Internacional Program (National Science Foundation) y el Fonacit.

permitir la suspensión del procesamiento normal a favor de una revisión periódica de las entradas, y el subsiguiente re-inicio del procesamiento, presumiblemente desde el punto donde fue parado.

- Reactividad basada en el conocimiento: un segundo aspecto de la reactividad está íntimamente enlazada a la estructura de conocimiento en la base de conocimiento del sistema. Si el conocimiento del sistema está altamente optimizado y compilado juntos, por ejemplo, a las reglas condición acción, el sistema puede reaccionar a entradas en un mínimo tiempo con una gran oportunidad de éxito debido a su respuesta oportuna (*timely response*). En una configuración extrema, estas reglas condición acción son guardadas en una tabla o en un arreglo que directamente relaciona las entradas de los sensores a las acciones dictadas a los efectores.

Típicamente, el comportamiento activo de una base de datos se describe usando reglas, las cuales tienen tres componentes: un evento, una condición y una acción. Una regla con tales componentes es conocida como reglas eventos-condición-acción o *reglas ECA*. El significado de estas reglas es: "cuando un **evento** ocurre, chequea la **condición** y si ésta condición se cumple, ejecuta la **acción**". Una regla permanece dormida hasta que se monitorea la ocurrencia de un evento, entonces la regla se 'dispara'. Las condiciones de la regla disparada son evaluadas, y si son todas verdaderas, la regla se agrega a un *modulo de conflicto*, en el cual se almacenan las reglas disparadas que son luego accedidas por el supervisor, quien selecciona la acción de la regla para su ejecución.

Las características de estructura y comportamiento de un Sistema de Bases de Datos Activa puede clasificarse, según [19], en tres áreas: modelo de conocimiento, modelo de ejecución, y modelo de administración.

El modelo de conocimiento representa la vista sintáctica de las reglas activas como es vista por el programador de las reglas. Esto tiene tres principales facetas, según [19]:

- *lenguaje de evento*: una notación con la que se especifican las condiciones que disparan una regla.
- *lenguaje de condición*: una notación usada para expresar restricciones adicionales las cuales pueden ser satisfechas por la base de datos después que la regla disparada es añadida al conjunto de conflicto.
- *Lenguaje de acción*: una notación usada para especificar el efecto que la regla debe tener cuando es ejecutada.

Como ya anunciamos, nos proponemos hacer un modelo formal de propósito general para un Sistema de Bases de Datos Activa, y estas notaciones que especifican eventos, condición y acción trataremos de integrarlas en un solo lenguaje con una sintaxis común y una semántica común. Un único lenguaje nos proporcionaría, de inmediato, la posibilidad de comparar modelos particulares que describan sistemas particulares, apelando al modelo general implícito en la semántica del lenguaje. Esto, desde luego, supone que hayamos podido crear ese modelo general de referencia para el lenguaje integrado, incorporando, en el lenguaje, todos los conceptos y constructos necesarios para describir todos los tipos y variantes de bases de datos activas existentes. Este es, desde luego, el mayor desafío.

Por su parte, el modelo de ejecución describe como las reglas interactúan en el contexto de todo el sistema de base de datos. Esto tiene los siguientes aspectos:

- *Granularidad de la transacción*: la naturaleza de los enlaces entre la ocurrencia de los eventos y la activación de las reglas – es posible que un evento individual ocurra cuando se dispare una regla, o que una colección de ocurrencia de un evento dispare una regla.
- *Modo acoplamiento*: la relación temporal y causal entre el disparo y la ejecución – por ejemplo, es posible que la acción de la regla sea ejecutada tan pronto como es posible después de la evaluación de la condición de la regla (inmediato), o en un punto retardado, tal como al final de la transacción (diferido).
- *Esquema de prioridades*: un ordenamiento especificando cuales reglas son consideradas primero cuando varias han sido disparada al mismo tiempo (ejemplo: cuando el conjunto de conflicto tiene más de una regla).
- *Políticas de efecto Nulo*: los cuales permiten cambios internos en los estados intermedios a una transacción para que sea ignorada por las reglas del sistema.

Un error común en los modelos formales, como muchas especificaciones basadas en lógicas modales, es relegar el equivalente al modelo de ejecución del sistema a un segundo plano, concentrándose únicamente en la estructura conceptual, sin mayor reparo por la operacionalidad del modelo. En este proyecto, trataremos de aprovechar las experiencias en lógica computacional que elaborar la estructura conceptual tratando de responder, al mismo tiempo, preguntas fundamentales acerca de la operacionalidad del modelo.

Finalmente, el modelo de administración comprende propiedades de las reglas bases tomadas como un todo, también como las operaciones para modificar las reglas:

- *Operaciones con reglas*: algunas bases de datos activas permiten que las reglas sean creadas o removidas, activadas o desactivadas, durante la ejecución.
- *Terminación*: sí o no la ejecución de una regla dada debe alcanzar un estado final.
- *Confluencia*: sí (en general no determinístico) el conjunto de reglas define un estado final único dado cualquier contexto de iniciación
- *Modelo de Equivalencia*: una noción de equivalencia semántica entre conjuntos de reglas, el cual es un prerrequisito para la optimización, ya que permite evaluar y decantar reglas equivalentes.

En [19] se evalúan diferentes propuestas de Sistemas de Gestión de Base de Datos Activa en términos de esos tres modelos y se describen los propósitos específicos para la descripción formal del comportamiento activo, por lo cual ese trabajo es el predecesor principal del proyecto que acá se describe. Sin embargo, al igual que otros esfuerzos que explicamos a continuación, ese trabajo no alcanza a integrar, en un todo coherente, los tres modelos que se describieron. Aún en este punto inicial del proyecto, tenemos razones para creer que un modelo basado en agentes podría proveer ese espacio integrado coherente.

Otros intentos por caracterizar las Bases de Datos Activas son:

- **Starburst in notación semántica**. El modelo denotacional del sistema de reglas activas Starburst [26] establece como puede especificarse formalmente el modelo de ejecución de un sistema de base de

datos activa existente. La especificación enfoca hasta el modelo de ejecución, y se concentra sobre el lenguaje Starburst de condición/acción. Es por tanto un modelo parcial de una base de datos activa.

- **Un marco OO para especificar el sistema de reglas activas.** Una técnica formal usada para especificar la semántica de los diferentes sistemas de reglas fue Object-Z [21], una extensión orientada a objeto del lenguaje de especificación formal Z. Como en la especificación de Starburst de la sección anterior, los detalles de los lenguajes de evento, condición y acción no están incluidos en las especificaciones, por lo que, como modelo, es también parcial.
- **Heraclitus: la ejecución de un lenguaje de descripción de modelo.** Hull y Jacobs [17] describen un lenguaje de programación de base de datos, *Heraclitus*, que es usado para modelar constructos de bases de datos activas. Lamentablemente, Heraclitus no incluye una especificación del razonamiento formal que se puede realizar sobre el lenguaje.
- **Un marco lógico usando Cálculo de Eventos .** El Cálculo de Eventos, de Sergot y Kowalski [18], es una especificación formal que establece que características del funcionamiento activo pueden ser capturadas en descripciones en lógica de primer orden [15]. La especificación del Cálculo no incluye, las características del modelo de ejecución, que se suponen heredadas de los mecanismos de razonamiento lógico. Es, por tanto, un modelo parcial, por lo menos hasta tanto no se definan esos mecanismos de razonamiento.
- **Una semántica operacional para el análisis de reglas bases.** La técnica de [28] proporciona una semántica operacional para la expresión del sistema de reglas usando un sencillo lenguaje de reglas condición-acción. La limitación del modelo es que se enfoca en el lenguaje de reglas de condición-acción únicamente.
- **Aplicación del Álgebra Relacional al análisis de reglas y optimización.** La especificación formal usando Álgebra Relacional soporta análisis [3] y optimización [4] de las reglas condición-acción. Sin embargo, en estos trabajos no se consideraron las características del modelo de ejecución y los eventos en la especificación formal.
- **Descripción de eventos usando Redes de Petri .** En [16] se presenta una descripción formal, abstracta y fácilmente implementable de un lenguaje de descripción de evento. La técnica usada para la descripción del modelo formal fue las redes de Petri coloreadas. El modelo no especifica las características de otro comportamiento activo diferente a los eventos, tales como un lenguaje de condición/acción, por lo que el modelo es también incompleto.

Las referencias anteriores sirven para ilustrar los esfuerzos de la comunidad científica para proveerse un modelo formal que describa las funcionalidades de los Sistemas de Bases de Datos Activa. Sin embargo, los modelos formales que han sido desarrollados no contemplan toda la variedad conceptual, menos aún, el rango de facilidades que caracterizan a dichos sistemas. Nuestro propósito es proponer un modelo formal que incluya las características ya descritas de estructura y comportamiento de un Sistema de Bases de Datos Activa. Para ello, en este trabajo apelaremos a una herramienta conceptual cada vez más usada para describir sistemas dinámicos complejos: Los Sistemas Multi-Agentes.

Introducción a la teoría de agentes. Un Agente es una entidad que puede *percibir* su medio ambiente, puede *asimilar* esas percepciones colocándolas en un dispositivo de memoria, puede *razonar* la información almacenada en ese dispositivo de memoria, puede adoptar creencias, *metas*, e intenciones por sí mismo y puede, de manera activa, captar los logros de esas intenciones, con un apropiado *control* de sus *efectores* [12]. Hemos descrito a un *Sistema Manejador de Bases de Datos* como un sistema de software cuyo propósito es definir, construir, mantener y operar, confiable y eficientemente, bases de datos multi-usuarios. Esta definición del Sistema Manejador de Bases de Datos nos permite relacionarlo preliminarmente con los Agentes describiendo, al Sistema Manejador de Base de Datos como un agente de software cuya meta principal es, por ejemplo, cuidar por el mantenimiento e integridad de una Base de Datos. Otras metas, como el atender las consultas a la base de datos, también podrían plantearse en escenarios en donde la propia Base de Datos haya sido enriquecida con otras fuentes de conocimiento. El agente podría fungir, por ejemplo, como gestor de ontologías asociadas a la BD o como almacén de estrategias óptimas para recuperar datos.

En cualquier caso, pretendemos utilizar la visión de Agentes de Software para definir Sistemas de Bases de Datos Activa. Con su esencia dinámica, los agentes serían, en esos casos, una extensión de los modelos de bases de datos activas construidos hasta la fecha. Sin embargo, hay todavía muchos elementos a contemplar en la integración de los agentes con las bases de datos. Uno de esos aspectos, que se ha tornado muy popular, tiene que ver con la integración de agentes con almacenes de información “semi o no estructurada” como la que ocurre en Internet. El proyecto emblemático en esa dirección se conoce como la Web Semántica [30].

Agentes, Ontologías y la Web semántica. Una de las metas de la Web semántica es la de hacer accesible la información basada en la Web al procesamiento de la máquina, donde habitualmente se requiera mediación humana. Una visión de la tecnología de agentes es precisamente la que es concerniente con la creación de procesos computacionales autónomos que minimizan o aumentan el comportamiento humano.

El objetivo trascendente que persigue la Web Semántica se puede resumir en esto: aprovechar el marcado (“tagging”) de textos para proveer esa información no estructurada con indicadores de significado y, así, poder desarrollar fuentes interconectadas más ricas en conocimientos en Internet.

Las notaciones básicas definidas por el consorcio W3C para la representación de la data en la Web semántica son el Marco de Descripción de Recursos (Resource Description Framework o RDF) y el Lenguaje Web Ontológico (Ontology Web Language u OWL).

RDF. El RDF [29] se define como un modelo simple en el cual los datos se refieren a RECURSOS y corresponde con un grafo dirigido y etiquetado. Las aristas en el grafo son etiquetadas con un símbolo del predicado, así el grafo puede también ser pensado como un conjunto de tres tuplas o *triples*:

Nodo-sujeto - símbolo del predicado - nodo-objeto

En RDF, los nodos representan los recursos o la data *literal*. Un recurso es típicamente identificado como un Identificador de Recursos Uniforme (Uniform Resource Identifier o URI). Un común tipo de URI es el URL [http](http://ejm) (ejm, <http://www.hp.com>), pero existen otros tipos. Un literal es un valor de data concreto, tal como un entero o cadena (string).

OWL. OWL es el lenguaje ontológico en el conjunto de especificaciones de W3C. OWL extiende RDF con una terminología específica declarando axiomas que definen clases, expresiones de clases y propiedades en una descripción lógica muy rica. La descripción lógica es un subconjunto del cálculo de predicado. Detalles acerca de las capacidades de representación de OWL están disponibles en el sitio Web de W3C [30].

En [31] se describen cuatro puntos genéricos en los cuales la tecnología de la Web semántica puede ser dirigida para beneficiar a los diseñadores de agentes. Estas son:

- Creación de vocabularios controlados. Un *vocabulario controlado* es un conjunto de símbolos (nombres de clases, nombres de predicados, etc.) que son permisibles para una aplicación dada. Simplemente escribiendo el dominio ontológico como un documento OWL ayuda a definir un vocabulario controlado, esto permite a los desarrolladores individuales de agentes chequear que los símbolos provenientes de los vocabularios controlados se usen correctamente.
- Uso de data semi-estructurada. Aplicaciones RDF usan representaciones construidas desde grafos de nodos enlazados. Esta codificación alguna veces es llamada *semi-estructurada*, porque tienen una estructura definida, diferente al lenguaje de texto natural, pero no es rígida como, por ejemplo, el esquema de una base de datos relacional. En un conjunto de datos semi-estructurados se puede añadir un nuevo nodo para codificar una nueva condición en esos datos o en el problema. Según como el agente que procesa data sea capaz de interpretar el significado de los nodos y la etiqueta de predicado, esta flexibilidad puede ser libremente explotada.
- Uso de consultas RDF. Existen un número de lenguaje especializados para realizar consultar a fuentes de conocimiento RDF. Esto permite hacer diseños para el manejo de consultas en una plataforma de agentes. Actualmente, los esfuerzos se están centrando para definir un lenguaje de consulta estándar [32].
- Uso de inferencias OWL. El modelo de inferencia usa la semántica OWL para manipular las restricciones que se deducen de la ontología y prioriza las afirmaciones, además de determinar si existen ontologías relacionadas con la que se manipula. El agente sería el responsable de las inferencias.

Esos recursos expresivos de la Web Semántica pueden resultar útiles para describir variantes de las Bases de Datos Activas, especialmente si resulta posible explicarlos con el mismo modelo general basado en Agentes que nos proponemos desarrollar. Así, un aspecto del proyecto de investigación que proponemos es el de un Sistema de Base de Datos Basado en Agente, que admitirá datos ambientales y que deben poder compartirse e intercambiarse a través de la Web.

Debemos dedicar un último aparte de esta propuesta a la presentación del dominio de aplicación seleccionado para el proyecto y de su justificación.

Bases de Datos Ambientales. La base de datos que queremos especificar debe manejar datos y metadatos ambientales. La definición básica de metadatos y que al parecer es aceptada universalmente es que *Metadato es dato sobre los datos*. Otras definiciones referidas en [25]:

“Nivel superior de la información, o instrucciones que describen el contenido, contexto, calidad, estructura y accesibilidad de una colección de datos específica.”

“Todos los datos y conocimientos que proporcionan información acerca de procesos, datos y entidades.”

En [25] definen a los sistemas de información para datos ecológicos, biológicos y ambientales como Sistemas de Información Ambiental; y añaden que los Sistemas de Información Ambientales permiten llevar a cabo estudios que abarcan periodos temporales grandes y espacios geográficos extensos, así como usar los datos para propósitos diferentes a los originales, por parte de personas distintas a las que recolectaron dichos datos. Los científicos que manejan datos sobre un aspecto particular de las ciencias ambientales, o sobre un lugar geográfico determinado, necesitan intercambiar datos entre sí debido a que los ecosistemas, climas y ambientes del mundo en realidad están estrechamente relacionados e interconectados y forman un sistema global. Las Bases de Datos Ambientales, entonces, contienen datos ambientales (datos meteorológicos, climáticos, vegetación, fauna, hídrico, suelos, población, grupos culturales, etc.); y estos datos deben poder compartirse e intercambiarse. Para compartir e intercambiar datos ambientales a través de ambientes heterogéneos, se deben establecer estándares y lenguajes de manejo e intercambio de datos y metadatos.

La estandarización de términos y un sistema unificado que describa los usos representan un beneficio importante para compartir información. Un ejemplo de estándar lo tenemos en el modelo de data relacional propuesto por Cook para la descripción de data botánica. El Economic Botany Data Standard (EBDC) [10] es un estándar para la descripción de los usos de plantas (en su contexto cultural), usando términos y descriptores estandarizados, y unido a conjunto de datos taxonómicos. Los datos, de esta manera se organizan en varios niveles, según la clasificación hecha por Cook.

En nuestro trabajo pretendemos proponer, como un sub-componente del modelo integrado de Base de Datos Activa, un estándar para la descripción de los datos y el lenguaje para el manejo e intercambio de datos y metadatos ambientales de manera que sea posible el compartir la información entre los diferentes centros de investigación.

Creemos que un Sistema de Base de Datos basado en Agente podría ser la solución para la administración de esta intrincada estructura de datos y metadatos que es organizada en múltiple niveles. La organización multinivel de los datos se establece mediante múltiples enlaces entre los mismos datos. La actualización de esta estructura requiere de consultas que asocien estos múltiples enlaces y construyan las respuestas adecuadas.

Recordemos que estos datos deben poder compartirse e intercambiarse mediante un Sistema de Información de Red. Esta acción de compartir e intercambiar datos puede dar origen a nuevos datos que se originen en un sitio y se necesiten llevar a otros sitios. Un Agente tendría como propósito la búsqueda de nuevas estructuras en los diferentes sitios que comparten información para enriquecer la Base de Datos y que mantengan los enlaces entre los datos.

Los datos ambientales son de naturaleza espacial: se representan normalmente en sistemas de información geográfica, SIG, a partir de objetos geométricos básicos como los puntos, segmentos rectilíneos, triángulos y otros polígonos de dos dimensiones y los cilindros, esferas, paralelepípedos y otros

poliedros en tres dimensiones. El soporte de los datos espaciales en la base de datos es importante para la administración eficiente de los datos en posiciones espaciales. Así que uno de los desafíos que se plantea este proyecto es explorar la utilidad de los agentes para efectos de representar, almacenar y manipular datos espaciales.

Otro punto importante en el tema de las Bases de Datos Ambiental es que ellas se utilizan para mantener un registro completo de la data ambiental. Esto es, vegetación y sus tipos, recursos hídricos, fauna, flora, población y grupos culturales que la habitan. Grupos de Investigación en esta área requieren de herramientas para simular tanto el comportamiento de los diversos entes que conviven en este ambiente y su influencia sobre el mismo, como el de entes externos al hábitat (por ejemplo, ordenanzas donde clasifican los usos de la tierra), que con su comportamiento y decisiones afectan dicho ambiente; para utilizarla como soporte para la toma de decisiones. Por esto la Base de Datos debe estar en capacidad de reaccionar ante estos estímulos internos como estímulos externos y proporcionar respuestas, operando concertadamente con los sistemas de simulación que operen sobre modelos de esas dinámicas ambientales. Así que la integración de manejadores de bases de datos con los sistemas de simulación es otro de los grandes desafíos que se plantea este proyecto.

El problema de la tesis. Pretendemos desarrollar un modelo formal de propósito general para un Sistema de Base de Datos Activa orientado a Agentes (preliminarmente llamado SBDAA). El SBDAA debería contemplar el rango de facilidades que soportan los formalismos y sistemas de bases de datos activas existentes, incluyendo los que hemos mencionado. Esperamos proponer un único lenguaje cuya semántica integre agentes, reglas, eventos, condiciones de operación y que dé cuenta de las diversas modalidades de almacenamiento de la información, desde almacenes estáticos, hasta servicios dinamizados por agentes sobre data semi-estructurada.

El modelo debe admitir data compleja. Específicamente queremos implementar el modelo en prototipos funcionales de bases de datos y metadatos científicos ambientales, introduciendo en nuestro modelo formal los constructos necesarios para su caracterización. Con el modelo formal esperamos también poder estudiar la relación entre las Bases de Datos Activa y los sistemas basados en ontologías que caracterizan a la Web Semántica, así como los simuladores de sistemas complejos.

Referencias

1. ACT-NET Consortium. The Active Database Management System Manifiesto: A Rulebase of ADBMS Features. In SIGMOF Record, vol 25, no. 3, September 1996.
2. Aiken, A., Widom, J. and Hellerstein, J.M. Behavior of database production rules: Termination, confluence, and observable determinism. In *ACM SIGMOD*, volume 21, pp,59-68, 1992.
3. Baralis, E., Widom, J. An algebraic approach to rule analysis in expert database systems. In J. Bocca, M. Jarke, and C. Zaniolo, editors, Proc 20th VLDB, pp,475-486. Morgan-Kaufmann, 1994.
4. Baralis, E., Widom, J. Using delta relations to optimize condition evaluation in active database. Technical Report Stan-CS-93-1495, Department of Computer Science, Stanford University, 1993.
5. Bradshaw, J., An introduction to software agents. In *Software Agents*, J. Bradshaw, Ed AAI Press, Menlo Park, California, USA, 1997, pp,3-46.
6. Campin, J., Paton, N.W., and Williams, M.H. A structured Specification of an Active Database System. *Information and Software Technology*, 37(1);47-51,1995
7. Campin, J., Paton, N.W., and Williams, M.H. Specifying Active Database Systems in an Object-Oriented Framework. *Submitted for publication*,1995.

8. Chamberlin, Astrahan, y otros. A History and Evaluation of System R. In Stonebraker and Hellerstein, editors, *Readings in Database Systems*, pp 54-68. Morgan Kaufmann Publishers, Third Edition, 1998.
9. Codd, E.F. A Relational Model of Data for Large Shared Data Banks. In Stonebraker and Hellerstein, editors, *Readings in Database Systems*, pp 5-15. Morgan Kaufmann Publishers, Third Edition, 1998.
10. Cook, F.E.M. Economic Botany Data Collection Standard. Prepared for the International Working Group on Taxonomic Databases for Plant Sciences (TDWG). Kew: Royal Botanic Gardens, Kew. En www.rbkew.org.uk. 1995.
11. Date, C.J., *Introducción a los Sistemas de Bases de Datos*. Prentice-Hall. Séptima Edición. 2001.
12. Dávila, J. A. Agents in Logic Programming. PhD thesis, Imperial College of Science, Technology and Medicine, London, UK, June 1997.
13. Dennet, D., *The intentional stance*. The MIT Press, Cambridge, MA, 1987.
14. Elmasri, R., Navathe, S. *Fundamentals of Database Systems*. Addison-Wesley. Third Edition, 2000.
15. Fernandes, A.A.A., Williams, M.H., and Paton, N.W. A logic-Based Integration of Active and Deductive Database, 1994. Submitted for publication.
16. Galzin, S. and Dittrich, K.R. Events in an active object-oriented database. In N.W. Paton and M.H. Williams, editors, *Rules in Database Systems*, pp 23-39. Springer-Verlag, 1994.
17. Hull, R. and Jacobs, D. Language constructs for programming active database. In Camps, R., Lohman, G.M., Semadas, A. editor, *Intl. Conf on Very Large Data Base*, volumen 17, pp 455-467. Morgan Kaufmann 1991.
18. Kowalski, R. Database updates in the event calculus. *Journal of Logic Programming*, 12:121-146, 1992.
19. Paton, N.W., Camping, J., Fernandes, A. y Williams, M.H. Formal Specification of Active Database Functionality: A Survey. In T. Sellis, editor *Proc 2^{do}. Int. Workshop on Rules in Database Systems*, pp, 21-35. Springer-Verlag, 1995.
20. Paton, N.W., Diaz, O., Williams, M.H., Camping, J., Dimn, A., y Jaime, A. Dimension of active behaviour. In N.W. Paton and M.H. Williams, editors, *Proc 1st. Int. Workshop on Rules in Database Systems*, pp, 40-57. Springer-Verlag, 1994.
21. Rose, G.A. Object-Z. In Stepacy, S., Barden, R., and Cooper David, editors. *Object-Orientation in Z*, pp, 59-77. Springer-Verlag, 1992.
22. Stonebraker, M., Hellerstein, J. Objects in Database. In Stonebraker and Hellerstein, editors, *Readings in Database Systems*, pp 461-465. Morgan Kaufmann Publishers, Third Edition, 1998.
23. Stonebraker, M., Jhingran, A., Goh, J. and Potamianos, S. On rules, procedures, caching and views in database systems. *In Proc. ACM SIGMOD*, pp 281-290. 1990.
24. Stonebraker, Michael. Retrospection on a Database System. In Stonebraker and Hellerstein, editors, *Readings in Database Systems*, pp 69-76. Morgan Kaufmann Publishers, Third Edition, 1998.
25. Torrén, R., *Desarrollo de Sistemas de Información Bio-climática*. Tesis de Maestría, Universidad de Los Andes, Facultad de Ingeniería, Mérida, Venezuela. 2003.
26. Widom, J. A Denotational Semantics for the Starburst Production Rule Language. *ACM SIGMOD Record*, 21(3):4-9, 1992.
27. Widom, J. and Finkelstein, S.J. Set-Oriented Production Rules in Relational Database Systems. *In Proceedings of the ACM SIGMOD Record International Conference on Management of Data*, pp, 259-270, 1990.
28. Zhou, Y., Hsu, M. A theory for rule triggering systems. In F. Bancilhos and et al., editors, *Proc. Extending Database Technology (EDBT)*, pp, 407-421. Springer-Verlag. 1990.
29. World Wide Web Consortium (W3C). The Resource Description Framework (RDF). 2004. <http://www.w3.org/RDF/>
30. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., and Stein, L. Owl Web Ontology Language Reference. W3C. 2004.
31. Dickinson, Ian. The Semantic Web and Software Agents: Partners, or just Neighbours?.
32. W3C. RDF Data Access Working Group. 2004. <http://www.w3.org/2001/sw/DataAccess/>
33. Uzcátegui, Mayerlin. Simulando Biocomplejidad Ambiental. http://ing.uvm.edu.ve/expotecnologia/curriculum/p_mayerling.pdf. Fecha consulta: septiembre 2005.