



Tecnura

ISSN: 0123-921X

tecnura@udistrital.edu.co

Universidad Distrital Francisco José de Caldas
Colombia

TREJOS BURITICÁ, OMAR IVAN

Consideraciones sobre la evolución del pensamiento a partir de los paradigmas de programación de computadores

Tecnura, vol. 16, núm. 32, abril-junio, 2012, pp. 68-83

Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257024143007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Consideraciones sobre la evolución del pensamiento a partir de los paradigmas de programación de computadores

Considerations on the evolution of thought from the computer programming paradigms

OMAR IVAN TREJOS BURITICÁ

Ingeniero en Sistemas, doctor en Ciencias de la Educación. Docente de la Universidad Tecnológica de Pereira. Pereira, Colombia. omartrejos@utp.edu.co

Clasificación del artículo: Reflexión (Conciencias)

Fecha de recepción: Marzo 12 de 2011

Fecha de aceptación: Noviembre 28 de 2011

Palabras clave: evolución, paradigmas de programación, pensamiento humano, tecnología.

key words: evolution, programming paradigms, human thinking, technology.

RESUMEN

El presente artículo aborda el tema de los paradigmas de programación explicándolos de una manera simple y presentándolos en una línea de tiempo conceptual que puede interpretarse como una necesidad en la evolución del pensamiento humano. Se hace una breve descripción, no solo de los paradigmas sino de sus aplicaciones, que a la postre también se han asimilado y entendido como paradigmas y, se presentan algunas reflexiones acerca de la incidencia que ha tenido la evolución del pensamiento en conjunto con las

nuevas necesidades del ser humano como de su proyección en el tiempo y lo que podría sobrevenir en el mundo de la programación de computadores en un futuro no muy lejano.

ABSTRACT

This paper discusses the programming paradigms of explaining them in a simple manner and presenting them in a conceptual time line can be interpreted as a need in the evolution of human thought. A brief description of paradigms not

only their applications, but in the end, it has also been assimilated and understood as paradigms and some thoughts presented about are not only the impact that has been the evolution of thought

in conjunction with the new needs of a man except the projection in time and what might occur in the world of computer programming in the not too distant future.

* * *

1. INTRODUCCIÓN

El presente artículo es producto del proyecto de investigación “Aplicación del Modelo 4Q de preferencias de pensamiento en el perfilamiento de los estudiantes de primer semestre de Ingeniería de Sistemas de la UTP y en el ajuste de la metodología de enseñanza en beneficio del aprendizaje”, que fue aprobado por la Vicerrectoría de Investigaciones de la Universidad Tecnológica de Pereira bajo el código interno 6-10-6, en el cual se busca aprovechar el conocimiento que se tiene del modelo 4Q de preferencias de pensamiento formulado por el doctor Ned Herrmann en el proceso de perfilar a los estudiantes de 1º semestre de Ingeniería de Sistemas de la misma universidad y, a partir de allí, encontrar elementos que permitan mejorar los resultados del aprendizaje.

La acertada concepción de lo que es un paradigma de programación, el estudio de los paradigmas de programación y un breve análisis de las necesidades que han surgido en la medida en que ha evolucionado el pensamiento humano, así como la relación entre los tres, representa un importante tema de revisión, dado que ni está desligada la evolución del pensamiento de la generación de nuevos paradigmas de programación, ni estos son una creación totalmente nueva e inconexa del desarrollo humano, pues se deben, entre otros factores, a la misma evolución humana. Detrás de este análisis se sobreviene una concepción, por lo menos aproximada, no solo de los paradigmas de programación sino de las estrategias y técnicas que, de una u otra forma, se han aceptado también como paradigmas en sí mismos, es decir, como formas específicas de resolver determinado con-

junto de problemas a partir de unos elementos y características comunes.

Durante mucho tiempo se ha promovido el estudio de la programación de computadores, desde 1945, cuando se inicio este mundo informático y tecnológico que hoy ha penetrado cada uno de los poros de la sociedad y de la vida. Desligarnos de la informática y de la tecnología en el mundo moderno sería tan fácil como convencernos de que la tierra es completamente plana. El mundo de hoy está atado, por sus mismas necesidades, a la tecnología informática y es por ello que tener claro los paradigmas de programación y sus vertientes constituye una base para que los profesionales de hoy, y especialmente los ingenieros de sistemas, puedan concebir el mundo tecnológico en el cual viven.

Establecer relaciones entre la evolución del pensamiento y los paradigmas de programación permitirá encontrar algunas relaciones causales entre cada uno de los paradigmas y la manera como estos han ido mutando en virtud de los nuevos problemas que han ido apareciendo y en los cuales la tecnología ha permitido resolver. Cuando se conocen los antecedentes de procesos de orden tecnológico, entonces se hace más fácil pensar prospectivamente y mirar hacia el futuro con el ánimo de saber qué viene en camino y cómo nos debemos preparar desde la Ingeniería de Sistemas para esas nuevas fronteras de conocimiento.

A manera de hipótesis, podríamos decir dos cosas: primero, que es posible establecer una relación entre la evolución del pensamiento humano a través de su historia con los paradigmas de programación y también que, basados en dicha relación,

es posible mirar prospectivamente hacia dónde avanza la programación y qué nuevos retos nos esperan. Es claro que el espíritu que motiva este artículo sostiene que en realidad no solo existe tal relación, sino que los paradigmas son consecuencia de dicha evolución. Una de las dificultades que presentan los estudiantes de programación, en sus primeros contactos con el área, radica en el hecho de no poder tener un panorama claro que los aproxime a una concepción general de los diferentes paradigmas y sus respectivas vertientes, pues normalmente se les sirven asignaturas de programación, enmarcadas dentro de las teorías y modelos pertinentes, pero pocas veces se les hace este proceso de familiarización con las diferentes formas de resolver problemas computacionales.

Se pretende con este artículo dejar una idea muy clara en los estudiantes, ¿qué es un paradigma de programación? y de lo que ha representado cada uno, así como de las herramientas conceptuales que tienen para resolver determinados problemas. Este artículo no va hacia lo puramente matemático, dado que dicho tema corresponde a la razón de ser de otro artículo.

Si bien no es este un análisis detallado de cada uno de los paradigmas, sí es una aproximación informal a las características y herramientas conceptuales con los que cada uno de ellos posibilita la solución de problemas computacionales. En todo el artículo está presente la opinión del autor y es por ello que, frente al contenido del mismo, muchos conceptos estarán basados en la experiencia más que en la formalidad de cada modelo. El esquema que sigue el presente artículo comienza por describir el concepto de lo que es un paradigma en algunas de las diferentes interpretaciones que podemos encontrar, luego define lo que significa un paradigma de programación, establece una relación con la evolución del pensamiento humano y continúa con las primeras formas que se usaron para resolver problemas computacionales. Se sigue con el modelo de la programación

orientada a objetos y con los lenguajes basados en matemáticas específicas.

Se habla de la programación que aprovecha y capitaliza las nuevas tecnologías y finalmente se plantean algunos elementos de juicio sobre las nuevas tendencias en programación. En el siguiente ítem se abre una discusión al respecto de los temas planteados y, al finalizar, se infieren unas conclusiones de todo el texto de este artículo, terminando con las referencias bibliográficas que han alimentado la base teórica del mismo.

2. ¿QUÉ ES UN PARADIGMA?

Etimológicamente hablando, la palabra “paradigma” viene del griego *παράδειγμα* que significa “ejemplo” o “modelo”. Esta primera aproximación nos permite conocer un perfil inicial no solo de lo que significa sino de hacia dónde apunta su orientación de significado. En términos generales, un paradigma puede considerarse como un modelo o un patrón que se acepta en una determinada disciplina científica o en cualquier otro contexto epistemológico [1].

Algunos autores como Kuhn definen la palabra “paradigma” como “Un conjunto completo de valores, creencias y técnicas que son compartidas por todos los miembros de una comunidad específica”. Esta definición nos hace pensar que, teniendo en cuenta la evolución del pensamiento y los cambios que en estos se van dando, un paradigma también puede concebirse como un modelo de acuerdo con el cual se construyen otros objetos. Tal es el caso de la gramática, como área de conocimiento, que concibe un paradigma como un tipo de flexión que puede nominar, pero también podría ser verbal o adjetiva y que llega a ser empleada como guía para el uso de otras palabras dentro de una misma clase. En el caso de la Lingüística, por ejemplo, un paradigma se concibe como un conjunto de unidades que pueden llegar

a cumplir el mismo rol dentro de un determinado contexto y que normalmente es sintáctico.

La retórica tiene una interpretación particular dado que como paradigma se entiende esa figura que persuade o disuade de una determinada acción. Todo esto nos lleva a pensar que un paradigma, en su más simple esencia, es una forma que se acepta culturalmente y a través de la cual se concibe o se interpreta el mundo. Esa forma de concebir constituye un conjunto de teorías acerca de una faceta de la realidad y con ello se ofrece la posibilidad de resolver determinados problemas compartiendo unos principios considerados fundamentales.

Técnicamente también se acepta que un paradigma es el término que se utiliza para denotar una filosofía específica de programación de computadores, es decir, una forma clara, concreta y definida de resolver problemas computacionales a partir de unos conceptos fundamentales. Por tanto, podríamos decir que en Programación un paradigma es un modelo o esquema fundamental que organiza nuestras opiniones con respecto a una determinada solución en particular y con ello se convierten en ese conjunto de reglas que rigen la disciplina de la Programación, aunque esta definición puede ser extrapolada a otras áreas del conocimiento.

Es de anotar que ese conjunto de reglas que aporta un paradigma, por lo menos por un tiempo, se consideran como unas verdades que no se cuestionan y que se aceptan y se reconocen universalmente. Además sirven para proporcionar modelos y formas de resolver problemas específicos que involucran una comunidad científica. En la evolución del pensamiento humano, los paradigmas han estado presentes en diferentes aristas de la concepción del mundo. La religión, la formación socioeconómica, las relaciones entre las sociedades y diferentes criterios de orden político, económico, social, cultural e, incluso, tecnológico han ido

marcando la manera como el ser humano concibe el mundo. Podría uno imaginarse que el hombre primitivo miraba a la naturaleza y a sus demás semejantes de una manera antes de conocer el fuego y la rueda y que esa mirada cambió drásticamente luego de conocerlos. ¿Qué podemos decir de la concepción que tenía el ser humano en épocas posteriores? Sencillamente algo similar, puesto que el mundo antes y después de la imprenta se concibió de manera diferente, asimismo, antes y después de la máquina de vapor, de la electricidad, de la electrónica, de la microelectrónica y de la moderna microbioelectrónica. Si bien es posible que las necesidades del ser humano fueran las mismas desde siempre, la evolución de la forma como él mismo concibe el mundo a partir de los avances que su creación le ha permitido, ha posibilitado nuevas y mejores soluciones. Aunque la mayoría de ellas han traído otros problemas, otras soluciones y, en esencia, otras formas de ver el mundo, lo cual nos pone ante un proceso de cambios de paradigma en la medida en que la sociedad avanza en el tiempo.

La concepción estética del mismo cuerpo y su asociación con el arte y otras expresiones de tipo cultural y social son muestra de que los paradigmas con que el ser humano concibe el mundo han cambiado profundamente. La misma tecnología electrónica ha proporcionado nuevas herramientas para que la vida del hombre, y su forma de vivirla, sea significativamente diferente. ¿Se adquiriría en estos tiempos un televisor sin control remoto? ¿Nos negaríamos a utilizar un cajero automático un día a las once de la noche y en un momento en el cual precisamos un dinero?

Todas esas son formas de ver el mundo que han ido cambiando a partir de las relaciones que se construyen entre los seres humanos y sus invenciones, sus avances y sus cambios en el conocimiento. De manera que un paradigma no es más que el producto de la misma evolución natural del ser humano y eso nos hace pensar que el área de

la Programación de Computadores no es la excepción, pues allí también la manera de resolver problemas computacionales ha ido cambiando, de una parte, por las mismas necesidades que van apareciendo y, de otra parte, por las posibilidades tecnológicas que el mismo ser humano va desarrollando.

Es posible que se hubieran querido tener unas interfaces gráficas hace cincuenta años cuando comenzó a desarrollarse la tecnología informática, pero posiblemente ni las necesidades lo exigían ni la tecnología lo posibilitaba. Por esa razón, los paradigmas de programación son un excelente ejemplo de la evolución del ser humano y cómo él ha afrontado los nuevos retos que van apareciendo. En este sentido también vale la pena hacer hincapié en que analizar y entender tal evolución nos puede dar pautas para mirar prospectivamente hacia el futuro y arriesgarnos a hacer algunas elucubraciones al respecto. Es, en este punto, donde vale la pena tener una pequeña definición de lo que significa programar, de su relación con las matemáticas y, de nuevo, de los nexos con el pensamiento humano y su evolución. Ese es precisamente el tema del siguiente ítem.

3. ¿QUÉ ES LA PROGRAMACIÓN DE COMPUTADORES?

En términos generales debe decirse que la programación de computadores es un camino para resolver un determinado conjunto de problemas a través de unos modelos y métodos específicos, ya que frente a los problemas que pueden resolverse con la programación de computadores siempre habrá posibilidades alternas que no necesariamente involucren tecnología.

Tal como se establece, la programación de computadores posibilita resolver un conjunto específico de problemas computacionales, es decir, aquellos en los cuales la solución necesita, según

el contexto, la participación de dispositivos de alta tecnología, o sea, aquellos que pueden recibir información, procesar, almacenar y elaborar respuestas. Los modelos y métodos con los que cuenta la programación de computadores son, precisamente, los que se derivan de los paradigmas de programación que no son otra cosa que una manera determinada de resolver un problema [2].

Por ejemplo, si tomáramos la vida de los hombres primitivos, podríamos decir que en aquellos tiempos transportar una cantidad de rocas implicaba la participación de la fuerza humana y de varios hombres. Esa era la forma de hacerlo en aquellos tiempos. Cuando aparece la rueda, es como si apareciera una visión diferente y cambian las posibilidades de resolver el problema de transportar una cantidad de rocas y, entonces, se pudo haber pensado en la construcción de un carromato que permitiera hacer más eficiente esta labor.

La programación de computadores, por definición, es el arte de construir programas a través de un conjunto de instrucciones que son entendibles y que pueden ser ejecutadas por el computador, como solución a un problema determinado. Programar implica conocer un problema, definir el objetivo por resolver, plantear su solución, escribir un algoritmo, codificar en un lenguaje de programación, hacer pruebas pertinentes, compilar y ejecutar el programa y, finalmente, evaluar los resultados que deben satisfacer los objetivos planteados. Esta es la más simple de las metodologías que puede ser transversal a todos los paradigmas debido a que no va hacia el problema en sí, sino hacia la solución como respuesta tecnológica. En este sentido ha de tenerse en cuenta que el presente artículo se referirá a la palabra “paradigma” en términos en que este es utilizado por la programación de computadores.

Es oportuno establecer que son las matemáticas las que han proporcionado la base para que la

programación de computadores sea una realidad y ello quiere decir que, finalmente, los paradigmas de programación no son más que la cristalización de un modelo matemático o de un conjunto específico de conceptos matemáticos. A todo paradigma de programación o cualquiera de sus vertientes subyace un modelo matemático. La presurosa vida tecnológica moderna deja pocos espacios para analizar la programación de computadores desde lo matemático y luego llevarlo a lo tecnológico. Adicionalmente, la seducción que ejerce la tecnología sobre los seres humanos en el mundo moderno ha ido logrando que se tienda más a conocer prontamente los elementos de la programación de computadores desde lo tecnológico que desde sus bases matemáticas.

Es esa misma matemática la que, poco a poco, ha ido llevando el pensamiento humano a instancias inimaginables en las cuales ha podido diseñar modelos y metodologías que le han permitido encontrar soluciones a los problemas. Esto quiere decir que tanto en la evolución del pensamiento del ser humano como en la evolución de los paradigmas de programación, han sido las matemáticas la gran base para que se pueda mutar ajustando las soluciones y encontrando nuevas, acorde con los cambios que va teniendo la humanidad en su devenir y en su existencia y en rima con las nuevas necesidades que van surgiendo bien por simple evolución o bien porque las nuevas soluciones también traen en sí mismas nuevos problemas.

4. LOS PRIMEROS PARADIGMAS

Construidos los primeros computadores, las necesidades comenzaron a aparecer. El contacto con la máquina fue directo y aún no habían aparecido los “mediadores” que hoy hacen mucho más cómoda la vida del programador. El concepto de “programación” no estaba tan definido como en la actualidad e incluso el mismo oficio de programador era bastante difuso, pues sencillamente cada

computador, del tamaño de una casa venía con las personas que lo manejaban, lo configuraban y podía sacarle el máximo provecho.

Como no eran de aplicación genérica, como los computadores de hoy, entonces tampoco era posible pensar en un abanico muy amplio de posibilidades y de soluciones a implementar a través de estas nuevas máquinas [3]. Casi que los “super computadores” (se usa el término *super* por su tamaño) tenía aplicaciones tan específicas que por esa razón no se pensaba, ni lejanamente, que un día estaríamos invadidos en nuestras casas y en todo nuestro entorno por estos aparatos.

El concepto de computador personal no se consideraba posible y los oficinistas eran tan específicos que había que pensar en su bienestar permanente para que, de pronto, esta estructura electrónica tan admirable no se fuera a quedar sin quien la operara. Son los tiempos prehistóricos de la informática y, lo sorprendente, es que esto sucedió apenas hace sesenta años, hacia 1945. Ese tipo de programación en la que unos hombres muy especializados tenían que activar y desactivar un tablero de más de veinte mil interruptores para lograr que el computador entregara los resultados esperados, se conoció como *Programación en Lenguaje de Máquina*.

Tiempos muy rudimentarios, las temperaturas eran muy altas y las instalaciones para operar los computadores quedaban dentro de las mismas máquinas. La relación de la sociedad con la tecnología era muy distante y solo algunas empresas, muy pudientes, podían tener este tipo de dispositivos, si el espacio se los permitía, para procesar su información. No existía la dependencia que hoy vivimos con las nuevas tecnologías y muchas operaciones manuales todavía eran válidas en aquellos tiempos. El ser humano concebía a la tecnología como esa aplicación de la ciencia que estaba permitiendo que procesos que duraban cinco años se pudieran procesar en tan solo seis

u ocho meses; algo que, comparativamente, era bastante favorable para las empresas y, especialmente, para los bancos. Procesar la información siempre fue una necesidad para la humanidad e igualmente poderla almacenar. Estos dispositivos gigantescos y que disipaban demasiado calor, terminaron cubriendo las necesidades que en ese momento iban abriendo paso.

Las empresas empezaron a pensar en la necesidad de ser más eficientes en el tratamiento de la información y la informática, como área de conocimiento, fue tomando forma poco a poco amén de dichas necesidades. El concepto de programa no existía en aquellos tiempos y la mayoría de las personas se sorprendía al saber que en determinada universidad norteamericana existía un aparato capaz de sumar con “excesiva” velocidad (velocidades que se medían en términos de segundos). Si bien la Programación en Lenguaje de Máquina no era un paradigma como tal, sirvió como un modelo para empezar a concebir la tecnología como parte de la solución de los problemas sin pensar en que un día dependeríamos tanto de ellas. Los modelos matemáticos asociados a esta forma de relacionarse con la tecnología estaban reducidos a las habilidades de algunos operadores en manejar un sistema numérico novedoso llamado Sistema Binario.

Este sistema tenía como base el número 2, los dígitos que lo conformaban eran el 1 y el 0 y la única operación que tenía era la suma. En comparación con un sistema tan complejo como el decimal cuya base era el número 10, dígitos del 0 al 9 y cinco operaciones (suma, resta, multiplicación, división y potenciación), el sistema binario era el llamado a simplificar muchos de los procesos que sucedían en el mundo real. Pero ¿cómo hacer que un sistema tan “complejo” como el sistema decimal pudiera encontrar equivalencias a todos sus procesos en un sistema tan “simple” como el sistema binario? Esa era la gran habilidad que tenían los técnicos y operadores de aquellos días

que eran tiempos en los cuales no se pensaba que un día, a nivel de las escuelas y colegios, se conocería, se apropiaría y se aplicaría el sistema binario con tanta facilidad como el decimal, tal como sucede en nuestros días.

La manipulación de un sistema como el binario en donde en vez de escribir el número 65 se escribe el número 01000001, trajo consigo riesgos de error que poco a poco se fueron cometiendo con las consecuencias a nivel de información y que tuvieron precios altos. Manejar grandes volúmenes de información representada en unos y ceros y saber que era tan fácil cometer errores, junto con el hecho de que cada operador o técnico de computadores en esos tiempos era casi “intocable” laboralmente, comenzó a hacer que el ser humano y la sociedad empresarial, particularmente, buscaran solución a tales malestares y fue entonces cuando se comenzó a pensar en que se necesitaba un mediador que hiciera un poco más fácil la relación con la estructura interna de esa máquina llamada computador.

En tales condiciones aparece el concepto de instrucción que no era más que una orden que podía ejecutar el computador y entonces las órdenes que, hasta el momento eran tan complejas, estaban dadas en unos y ceros, se volvieron más “digeribles”. Fue cuando apareció la posibilidad de construir pequeños programas basados en un lenguaje que se conoció como Lenguaje Ensamblador. A este tipo de programación se le conoció como *Programación a Bajo Nivel* pues de todas formas las instrucciones, a pesar de que no eran del todo entendibles, por lo menos lo eran mucho más que un conjunto de bits, nombre que se le puso a los unos y a los ceros.

La Programación a Bajo Nivel, como tal, tampoco se considera como un paradigma formal de programación, pero el hecho de que permitiera un cambio significativo en la relación técnica con la máquina por parte de los operadores (que ya se

empezaban a llamar como “programadores”) la convirtió en una forma de ver el mundo de esas nuevas tecnologías. La relación del ser humano y de la sociedad con la tecnología comenzaba a aproximarse y la idea de conocer un lenguaje como el Ensamblador y poder, a través de él, construir soluciones a diferentes problemas amplió el abanico de posibilidades de aplicación de los computadores y, con ello, también amplió su mercado, lo cual no le vino nada mal a las empresas fabricantes de estos maravillosos aparatos.

El ser humano comenzaba a ver que era posible resolver muchos nuevos problemas con el computador y que las instrucciones, finalmente, terminaban siendo manejables. Una orden como ST (que significaba “almacene”, del inglés storage) era mucho más entendible que 10001010 como pudo haber sido la orden original. El conjunto de órdenes (instrucciones) no era tan amplio como para no poderse manejar y se fue encontrando, poco a poco, más comodidad en las órdenes del Assembler que en los bits del Binario.

La especialización en el trabajo de programador, a pesar de que se mantuvo, por lo menos comenzó a tener más adeptos, pues estos ya tenían que lidiar con un entorno técnico más cómodo. La agrupación de instrucciones recibió un nombre que cambiaría por siempre la relación del hombre con la tecnología, se llamó Programa. De allí se derivaron dos términos: Programador y Programación. Como Programador se reconoció a la persona que se encargaba de pensar, diseñar, construir, probar y poner a funcionar un programa en un computador y como Programación se llamó el área de conocimiento que debía ser dominada por los programadores para hacer efectiva su labor.

El desarrollo de los programas, en aquellos tiempos obedeció al instinto propio de cada programador y, entonces, cuando uno de ellos faltaba por alguna razón, poder leer uno de sus programas, poder entenderlo y poder corregirlo o ajustarlo re-

sultaba ser una tarea muchas veces tan difícil que era más fácil volver a hacer todo, que corregir lo que ya estaba hecho. Esta realidad volvió a poner al ser humano de cara a un problema nuevo que debía resolver: ¿cómo construir una metodología que permitiera hacer un primer intento de unificar lo que entonces se llamaba: Lógica de Programación?

Es en ese momento cuando se comienza a pensar en el concepto de Paradigma de Programación, es decir, en una forma estándar de resolver los problemas en los cuales la solución se podía implementar a través de un computador y que además fuera aceptada por los profesionales del mundo de la programación de aquellos días. A través de un análisis matemático muy riguroso se llega a la conclusión de que todos los programas que se habían hecho hasta el momento se hubieran podido hacer basados en tres estructuras de pensamiento: las secuencias de instrucciones (que se resumían en el hecho de que cada instrucción se ejecuta solo después de la anterior y antes de la siguiente), los condicionales (se puede escoger uno de dos caminos lógicos cuya escogencia depende de una condición lógica) y los ciclos (se pueden reiterar varias instrucciones la cantidad de veces que una condición lógica lo permita).

Aparece el concepto de Lógica de Programación y con él viene de la mano el de Paradigma. En realidad lo que aparece en la escena es el primer paradigma cuyo nombre, debido a que se basa en unas estructuras de pensamiento, es *Paradigma Estructurado*. Este paradigma permitió ver en la programación la solución a muchos más problemas de los que hasta ese momento se habían resuelto. Se seguía aproximando más la relación entre el ser humano y la sociedad con el mundo de los computadores.

En este momento aparecen Lenguajes de Programación mucho más digeribles. La instrucción READ hace suponer que se le está indicando al

computador que lea algo, mucho más fácil que la instrucción RD o la posible instrucción 01100100. De esta manera la programación de computadores se hizo aún más alcanzable para muchas personas. Se hizo más seductor el oficio de programador y se empezó a pensar en que este oficio podía profesionalizarse.

El paradigma estructurado abrió las puertas para la *Programación Imperativa* que es un tipo de programación en la cual se le dan órdenes al computador y este las ejecuta literalmente. Los lenguajes de Programación alcanzaron un nivel altamente comercial y se inició el proceso de crear aplicaciones (programas) y venderlos como productos comerciales, algo que hasta el momento no era tan fácil. Esto llevó a que la estructura electrónica de un computador fuera mucho más pequeña, más cómoda, más económica y más alcanzable.

Muchos comenzaron a ver en el computador un nuevo electrodoméstico y, aunque algunos gerentes no lo creyeron, otros decidieron arriesgarse y terminaron tomándose por completo y para siempre, hasta ahora por lo menos, el mercado informático. ¿Qué pasó con las compañías fabricantes de electrónica que no pensaron que un día la sociedad estaría inundada de computadores? Sencillamente, desaparecieron porque el mercado informático es así de, insensiblemente, voraz. A todo esto subyace el intento del hombre de tratar de imitar la realidad a partir de los modelos que le posibilitaba la informática de aquellos tiempos, muy distante de los tiempos modernos, pero que cumplía con los requerimientos de la sociedad de esos días. No ha de olvidarse que desde esos días hasta los presentes solo han transcurrido cuarenta años.

El paradigma estructurado comenzó a resolver una muy buena cantidad de problemas pero, como todo en el ser humano, el mismo hombre comenzó a encontrar sus propios límites y comenzó a

pensar en que este paradigma tenía unas fronteras desde las cuales algunas cosas eran muy difíciles de imitar. Fue cuando comenzó a planear la forma de ampliar las soluciones que había encontrado para aproximarse mucho más a la realidad, el gran anhelo de la sociedad informática.

5. APROXIMÁNDOSE A LA REALIDAD

El espíritu de la programación de computadores radica en poder construir soluciones que simulen la realidad que vivimos por fuera del mundo electrónico y que estas soluciones puedan ser ejecutadas por computadores. Sobre esta premisa surge una interpretación muy interesante que generó una aproximación a la realidad tal como la vivimos. El mundo tangible está construido a partir de objetos tales como un vaso, una puerta, una mesa, un código, etc.

En su más simple descripción un objeto tiene unas características y unos usos, es decir, el objeto puede ser descrito a partir de sus atributos y está destinado para algunas funciones específicas. Por ejemplo, un vaso, tiene un peso, una altura, un material que lo conforma, una capacidad, unas medidas geométricas, una forma geométrica, una densidad en su material, una transparencia, una determinada capacidad de resistir al calor y otros atributos muy específicos.

En su uso, el vaso puede servir para contener un líquido, para hacer de pisapapel, para retener una puerta, para encerrar un insecto (si se invierte) y para otras tareas, todas ellas con una especificidad clara. Esto llevó a pensar que si el mundo estaba construido a partir de objetos y que dichos objetos tenían unos atributos y estaban destinados a unas funciones, entonces se podía simular esta situación en el computador. Efectivamente, fue este principio lo que llevó a que se formulara un paradigma que ha ido modificando poco a poco la relación de la sociedad con la tecnología y, parti-

cularmente, con el mundo de la programación. Se llamó *Programación Orientada a Objetos*.

Los objetos a los que hace referencia el nombre son un tipo de datos, que es lo que se maneja en la informática, y que están constituidos básicamente por atributos y métodos. Los atributos son las características del objeto que pueden ser definidas, manipuladas (a nivel de datos), almacenadas y modificadas. Los métodos son los procesos que se pueden asociar con dicho objeto y que también pueden ser configurados o “programados” dependiendo de las características que queramos conferirle a dicho objeto.

Por ejemplo, una ventana de las que se usan en cualquier versión de Windows tiene unos atributos y unos métodos. Una ventana tiene altura, ancho, color, botón, bordes, etc. Con una ventana uno puede ampliarla, reducirla, minimizarla, suprimirla, cerrarla, cambiarle el tamaño, ocultarla, etc. Podemos entonces decir que una ventana es un objeto de programación y efectivamente lo es.

Concebida así la programación se hizo mucho más fácil aproximarse al mundo real, pues en los paradigmas anteriores era necesario hacer una cantidad de re-interpretaciones para que ese mundo real tuviera una equivalencia en el mundo electrónico que posibilitaba la informática. Lo más interesante es que, con la POO (sigla que se ha acuñado para describir abreviadamente la Programación Orientada a Objetos) no solo se pudo describir el mundo tangible que nos rodea sino también el mundo intangible, el mundo de la imaginación, el mundo de los juegos, de las fantasías, de todas aquellas concepciones que el ser humano difícilmente había podido “construir” en el computador.

Con la POO aparecieron nuevos conceptos de programación tales como la herencia, el encapsulamiento y el polimorfismo que, a la luz del paradigma imperativo, eran conceptos difíciles de

implementar (aunque ya se han desarrollado mecanismos para simularlos). Estos conceptos nuevos fueron apareciendo, producto de las nuevas necesidades del ser humano y de ese intento franco que ha ido realizando el ser humano de aproximar el mundo informático al mundo electrónico de manera que este simule a aquel.

Sobre esta nueva forma de ver el mundo, a partir de objetos, fueron apareciendo nuevos retos tales como las posibles relaciones entre atributos y entre objetos y esa mezcla de conceptos fue generando el marco para que nuevas formas de pensamiento, basadas en la POO, comenzaran a tomar pista en el escenario de la programación. Los elementos comunicativos que incorporaba la programación junto con una proximidad mayor de la sociedad hacia la tecnología y hacia los dispositivos electrónicos hizo pensar en que “mostrar” el mundo real en la pantalla y hacer que el usuario se sintiera como si estuviera interactuando directamente con su propio mundo era una necesidad si bien no informática por lo menos comercial.

Es aquí donde, aprovechando los aportes de la POO, aparecen nuevos elementos visuales comunicativos e informáticos como los íconos y con ellos aparece todo un entorno que simplifica la interacción entre el ser humano y la máquina (usando el término que convencionalmente se sigue utilizando para el computador). Irrumpe en el mundo el concepto de ventanas y el mundo electrónico, que otrora se basaba solo en el texto, hoy tenía una frontera: los gráficos.

El mundo informático ahora sí, con este cambio, irrumpió en la vida de la sociedad y fue ahí cuando comenzó a penetrar cada uno de sus rincones. Derivada a la POO y con una contundencia tal que fue convirtiéndose en el modelo de resolución de problemas en lo que correspondía a su interfaz, apareció en el mundo de la programación lo que hoy se conoce como la Programación Visual que no es más que una forma de programar en la que

el programador (valga toda esta redundancia) tiene a su disposición una serie de objetos ya definidos y validados que hacen mucho más sencilla su labor y que le permiten que destine su tiempo más a los procesos y a su eficiencia que al diseño de la interfaz.

Los lenguajes visuales, tan cómodos para el programador, se tomaron rápidamente el mercado y sus programas, mucho más cómodos para el usuario, se convirtieron en el factor diferenciador de tiempos pasados y tiempos modernos. La idea de los íconos constituyó un elemento comunicativo de un poder inimaginable tanto que ello permitió que grandes fortunas económicas y verdaderos imperios se construyeran a partir de su incidencia y aceptación en la sociedad.

La programación visual es una muy buena derivación, y si se quiere aplicación, de la POO que ha permitido, ante todo, simular visualmente el mundo real para bien, comodidad y dependencia del usuario. La dependencia del ser humano por la tecnología se hizo realidad y tal entorno es lo que ha hecho que se disparen las ventas de computadores en el mundo, pues en últimas es como si el ser humano tuviera su entorno contenido en un portátil.

El insondable mundo del pensamiento humano fue creando sus nuevas necesidades y, en la búsqueda de hacer que cada vez la programación fuera más sencilla y más efectiva, se empezaron a perfilar tendencias que lo facilitarían. Es así como aparece una forma de pensar la programación que constituyó un paso adelante en el mundo de la informática. Aparece la *Programación Declarativa* como respuesta a este nuevo reto.

En la Programación Declarativa se condensan instrucciones de muy alto nivel, procesos completos reducidos a una sola orden con la cual el computador es capaz de interactuar sin mayores problemas, facilidades inimaginables por el ser

humano o que su concepción, hasta ese momento, involucraba un nivel de complejidad tal que exigía altas destrezas para su ejecución. La programación, otrora labor de grandes especialistas, estaba cada vez más a la mano de las personas del común, del ciudadano de a pie y sus resultados, los programas, eran más entendibles y mucho más fáciles de mantener.

Dentro de esta nueva forma de pensamiento que aprovechaba lo mejor de otros paradigmas, pero tenía sus propias características, se empiezan a generar vertientes que hacen más efectiva la programación y que mantienen la filosofía de lo declarativo. Es aquí donde la *Programación Funcional* irrumpe en lo técnico pues su fundamentación matemática ya se tenía hacía más de cincuenta años.

En la programación funcional la concepción general de solución a un problema parte de tres principios sencillos: la simplificación del objetivo, la reutilización del código y la reducción del tiempo en pruebas. La simplificación del objetivo se basa en la estructuración de una solución a partir de una unidad que es la que cambia el panorama y que se conoce con el nombre de función. Una función es una pequeña unidad que requiere unos argumentos para que cumpla un pequeño objetivo.

Una función en unión con otras funciones independientes e interdependientes son las llamadas a satisfacer el objetivo general. La labor del programador entonces va hacia los procesos y a generar una destreza que le permita identificar las funciones que necesita, a verificar si estas ya existen, construir las que no existan e hilarlas para que, conjuntamente, logren el objetivo propuesto. El reto es bastante innovador pero significativamente útil para resolver determinados problemas.

El cálculo Lambda desarrollado por Alonzo Church fue la base para que, muchos años después, la programación funcional fuera una realidad. La pro-

gramación funcional hace uso de diferentes recursos tomados de otros paradigmas, pero incorpora nuevos elementos para que la programación sea cada vez más fácil y más efectiva, espíritu principal de los cambios que se dan en las tendencias de la programación de computadores.

La construcción de programas es mucho más simple y el planteamiento de soluciones se hace más ágil y menos complejo. ¿En un panorama tan promisorio como este, qué nuevos horizontes tendría la programación de computadores? Ese es precisamente el tema de nuestras próximas secciones.

6. MODELACIÓN MATEMÁTICA

Basados en estas formas de pensar lógicamente, que al final es lo que va marcando el camino a partir de los paradigmas, y teniendo en cuenta todas las posibilidades que cada vez se iban abriendo con la tecnología y su participación en la solución de problemas de la sociedad, se empezó a pensar en la necesidad de rescatar, para bien de esa misma sociedad, modelos matemáticos que habían sido diseñados hace mucho tiempo y que solo hasta ese momento, como ha sucedido con muchos de los aportes de la matemática, se veían como camino de aplicación y aprovechamiento tecnológico.

Saltaron a la escena tres formas de programación dejando en claro que su modelación matemática existía desde hace mucho tiempo y que no necesariamente el presente artículo muestra un recorrido riguroso a nivel histórico de la evolución de los lenguajes de programación [4]. Se presenta la esencia de la lógica que ha permitido construir los paradigmas que hoy existen y su relación con la misma evolución del pensamiento.

La primera forma de programación de la que hablaremos es la *Programación Lógica*. Este tipo de programación lo podemos sintetizar de la si-

guiente forma: en un lenguaje tradicional (o sea aquel lenguaje de programación que se basa en paradigmas como el imperativo) la programación se reduce a la construcción de soluciones de problemas a partir de la disposición apropiada de una serie de instrucciones y sentencias. La buena disposición de las estructuras existentes junto con el aprovechamiento de los recursos que tiene el lenguaje determinado, permiten que la solución finalmente sea efectiva y cumpla con el objetivo de resolver el problema.

En la programación lógica se acude a una forma de trabajo descriptiva de manera que se puede establecer una relación entre entidades a través de lo cual se indique qué se va a hacer y no precisamente el cómo, que es la esencia del modelo imperativo. Según su creador, Robert Kowalski de la Universidad de Edimburgo, todo algoritmo se puede construir a partir de la especificación del conocimiento en un lenguaje de programación formal (que involucra la lógica de primer orden) y un problema que se quiera plantear se puede resolver a través de un mecanismo de inferencia (que procede como control) actuando sobre aquel.

Si bien la programación funcional puede considerarse como una evolución de la programación lógica, perteneciendo ambas al paradigma de programación declarativa, el orden de presentación de estas dos formas de programación ha sido basado en lo que ha representado en la evolución del pensamiento humano y no por razones de índole histórica.

El segundo modelo de programación, aceptado formalmente como otro paradigma de programación, lo constituye la *Programación por Restricciones*. En este paradigma las relaciones entre las variables son expresadas en los términos en que lo permitan las ecuaciones que se asocian con la solución. Sin duda, este es un paradigma que representa un interesante desarrollo en la programación de computadores y ha sido reconocido como

uno de los grandes pasos dados en la investigación en computación.

En este paradigma como su nombre lo indica, se especifican un conjunto de instrucciones, que deben ser satisfechas por cualquier vista que tenga la solución que se plantea al problema en lugar de establecer, de manera determinada, los pasos para llegar a la solución. Tiene una relación muy clara con la programación lógica al punto que un programa en programación lógica tiene una equivalencia directa en programación con restricciones y viceversa, lo cual se hace pensando en la eficiencia.

En su más simple esencia, en la programación con restricciones se busca un determinado estado en el cual se puedan satisfacer, al tiempo, una gran cantidad de restricciones. En este sentido se parte del principio de que un problema está definido como un estado específico de la realidad en donde existe un número determinado de variables con valores que no se conocen. Un programa que ha sido desarrollado a partir de la programación por restricciones, finalmente lo que hace es buscar los valores que satisfacen todas las variables.

La tercera forma de programación que aparece en escena corresponde a la *Programación Concurrente*. En este estilo de programación se le da prioridad a la ejecución simultánea de diferentes tareas interactivas. Dichas tareas bien pueden ser un conjunto de procesos o diferentes hilos de ejecución que hayan sido creados por un mismo programa. Las tareas mencionadas pueden ser ejecutadas en una sola unidad de proceso, pero también pueden ser ejecutadas en varios computadores (léase varios procesadores).

La programación concurrente enfatiza mucho más en la interacción entre tareas que en la especificación de proceso como tal. Por razones suficientemente entendibles, desde lo técnico, este estilo de programación (que algunos consideran

un paradigma informal) tiene una relación directa con la programación paralela que algunos autores llaman programación distribuida. Una acertada organización de la secuencia de interacciones o de procesos de comunicación entre las tareas junto con un acceso coordinado a los recursos que se están compartiendo resulta ser la clave para el planteamiento de soluciones eficientes.

En relación con el pensamiento humano y su nexo con la tecnología, estos modelos de programación pretenden aprovechar “nuevos” caminos teóricos para encontrar soluciones más eficientes y más efectivas a cierto conjunto de problemas de la sociedad. Esto ha llevado a asociar mucho más la programación lógica con la Inteligencia Artificial, la programación por restricciones con procesos matemáticos de alto nivel y la programación concurrente con redes y computación distribuida.

7. APROVECHANDO LAS NUEVAS TECNOLOGÍAS

Si la programación de computadores ha estado ligada a la evolución del pensamiento humano y si este ha tenido un nexo con la tecnología, es apenas lógico pensar que la aparición de nuevos medios de información y comunicación han incidido en la aparición de nuevas formas de resolver problemas que si bien parten de paradigmas existentes, de una u otra forma se han ido convirtiendo en modelos para resolver problemas de la sociedad y ello los ha llevado, de manera indirecta, a ponerlos en el nivel de nuevos paradigmas [5].

La alta penetración de Internet en todos los rincones de la sociedad y su posible aprovechamiento en muchas situaciones problemáticas de ella, ha llevado a que se acuñe el término Programación para la Web que no es más que la aplicación de los conceptos teóricos junto con las posibilida-

des tecnológicas de la Web para implementar soluciones eficientes con programación. En esta forma de resolver problemas es muy importante tanto el acceso a las Bases de Datos distribuidas como el aprovechamiento de los servicios telemáticos que, precisamente, son las principales características de este modelo de programación.

Es la misma Programación para la Web la que ha llevado, con la explosión comercial de los dispositivos móviles (léase aparatos celulares, iphone, ipad, etc.), a que se piense en un modelo muy específico, por sus características, y que se ha coincidido en llamar Programación para Dispositivos Móviles. En este modelo, considerado por algunos ya como un paradigma, se destaca fundamentalmente el aprovechamiento de los servicios inalámbricos de conexión y comunicación como base para el desarrollo de la programación.

Como se puede ver estas dos tendencias, consideradas por algunos como paradigmas, han sido el resultado de encontrar soluciones eficientes y prácticas en el mundo de los dispositivos modernos y, viviendo la gran penetración que estos han tenido en la sociedad, se hace imposible pensar que el campo de la programación no haya sido permeado por dichas innovaciones tecnológicas.

8. LAS NUEVAS TENDENCIAS

Acorde con lo expuesto anteriormente, las necesidades del ser humano son como su pensamiento, están en permanente evolución. Eso ha llevado a que en la actualidad se piensen en nuevos modelos que sigan, en ruta lógica, resolviendo los problemas que aparecen y que sus soluciones satisfagan no solo el instante que vivimos sino el contexto en el cual lo hacemos.

Tendencias modernas como la Programación Literaria, la Programación Inteligente, la Programa-

ción Virtual, la Programación para la Animación y la Programación 3D forman parte de ese mundo de posibilidades que pueden llegar a cristalizarse como paradigmas formales y que, actualmente, son solo planteamientos de cara a un inmediato futuro. La Programación Literaria aborda el problema del código como base primigenia para la generación de programas y, partiendo del principio de la documentación estándar, pretende excluirlo de forma que entre el programador y la máquina el nexo sea solamente la solución planteada en términos totales del programador y no como sucede hoy que dichas soluciones deben escribirse en términos de lenguajes de programación para poder ser ejecutados apropiadamente.

La programación inteligente busca resolver el problema de que los programas se modifiquen y se ajusten a ellos mismos de manera que en la medida en que los problemas que resuelven vayan mutando, ellos también lo vayan haciendo [6]. Tal vez sea una aparente utopía pero varios casos concretos prácticos han demostrado que esto es totalmente posible. El avance de la tecnología tendrá siempre elementos propios en su insondable mundo [7].

La programación virtual busca que los programas no sean unidades de código como tales, sino soluciones que, sin tener una presencia específica, puedan aportar una solución concreta a los problemas. No ha de olvidarse que estas tendencias solo están en proceso de “cocción” y que lo que ahora podría parecernos un planteamiento imposible, tal vez la misma tecnología nos lo desmienta de cara a un inmediato futuro.

La programación para la animación y la programación 3D buscan lograr el proceso de la imitación de la realidad en toda su plenitud, al punto de que el ser humano pueda interactuar, como ya lo está haciendo, con las creaciones que a partir de allí se hagan. La aparición de la tecnología MRI (Medical Resonance Image), los nuevos disposi-

tivos y conceptos como la Realidad Virtual y Realidad Aumentada han permitido que se abra una gran cantidad de posibilidades en esta línea que antes eran inimaginables.

Hemos de admitir que el ser humano cada vez tendrá más problemas nuevos para resolver, con más variantes y con más exigencias, pero afortunadamente es el mismo ser humano quien busca y resuelve dichos problemas acudiendo, como es este caso, al avance en la tecnología y a la investigación de punta.

9. DISCUSIÓN

Como se ha expuesto es natural que la evolución del pensamiento, y exactamente la relación del ser humano con la tecnología, ha ido siempre en el sentido de buscar nuevas y mejores soluciones a partir de los avances tecnológicos existentes. Naturaleza misma del pensamiento humano que ha buscado soluciones en el mundo que le rodea y aquellas soluciones que aún no son alcanzables, las ha buscado a partir de las fronteras de la ciencia.

Esta tendencia es la que ha hecho que la misma ciencia avance, que los problemas de la sociedad se resuelvan y que, de paso, aparezcan nuevos problemas. ¿Por qué no desaparecen entonces los problemas? Porque la naturaleza misma del hombre está en evolucionar que no es otra cosa que desarrollarse, pasando de un estado a otro, supuestamente mejor, pero por lo menos diferente. En el instinto del ser humano como mamífero racional está buscar siempre nuevos estados que sean mejores que los anteriores.

Esto nos lleva a pensar que el horizonte que tiene todavía la programación de computadores es bastante amplio y que, independiente de la mirada que podamos hacer prospectivamente en esta área, serán todavía muchas las necesidades que

quedarán por resolver y muchas más otras que faltan por aparecer. Con toda seguridad aparecerán nuevas tendencias y nuevos modelos a través de los cuales se pueda ver el mundo real, sus problemas y se puedan buscar sus soluciones. Otros paradigmas podrían estar a la vuelta de la esquina, pero por ahora pareciera que el mundo puede atenderse con los que ya existen.

La programación orientada a objetos, como paradigma, ha sido la que en la actualidad se acepta más, pues su aproximación a la realidad y a la interpretación de ella ha permitido construir soluciones que simulan dicha realidad, creando entornos virtuales frente a los cuales el usuario común pareciera estar en su mundo real. Tal vez estemos *ad portas* de que un día simplemente como usuarios enunciemos nuestra necesidad y entonces, de manera autónoma y automática, el computador podrá atendernos a través de una solución determinada.

Si este momento está cerca o no, no lo sabemos. Lo que sí podemos inferir es que en la medida en que el ser humano tenga nuevas necesidades, su propio pensamiento irá buscando, de la mano de la tecnología o de la ciencia o de cualquier otra frontera del conocimiento, soluciones que hagan su vida más agradable o, por lo menos, que tiendan a ello. Todo esto pareciera contradecirse con la industria armamentista y con el mundo de la guerra, pero reflexiones al respecto y evaluaciones de lo que ello ha significado será tema de otro artículo.

10. CONCLUSIONES

A manera de conclusión podemos decir que sigue siendo muy vigente el mundo de la programación de los computadores con sus soluciones y sus posibilidades, amén de las nacientes e insatisfechas necesidades del ser humano. Conocer los paradigmas nos permite saber el camino que ha recorrido

el pensamiento del hombre y confirmar, una vez más, que como mamífero racional nunca deja de buscar soluciones a los problemas.

Conocer dichos paradigmas, o por lo menos tener una idea clara de ellos, posibilita tener herramientas para realizar una mirada prospectiva y que, de una u otra forma, podamos saber hacia dónde tiende el mundo de las necesidades de la sociedad

y que nuevas puertas podrán estar próximas a ser abiertas. Cabe destacar el importantísimo papel de las matemáticas en el desarrollo y consolidación de los paradigmas de programación. Si bien este artículo no es ni un recorrido histórico plagado de fechas ni un análisis profundo de la evolución matemática, no se puede negar la importancia de tener en cuenta que a todo paradigma le subyace una teoría matemática que lo fortalece.

REFERENCIAS

- | | |
|--|---|
| <p>[1] B. Brendan, <i>A New Approach to Computer Science</i>, University of Massachussets, USA, [en línea]. Disponible en: http://dl.acm.org/citation.cfm?id=1059943</p> <p>[2] P. V. Roy and S. Haridi, <i>Concepts, Technics and Models of Computer Programming</i>, Swedish Institute of Computer Science, 2003.</p> <p>[3] M. Felleisen, R. B. Findler, M. Flatt and S. Krishnamurthi, <i>How to Design Program, an Introduction to Programming and Computing</i>, London: Massachussets Instituted of Technology, 2003.</p> | <p>[4] H. Eves, <i>An Introduction to the History of Mathematics</i>, USA: Saunders, 1990.</p> <p>[5] O. Buriticá, <i>Fundamentos de programación</i>, Manizales, Colombia: Papiro, 2004.</p> <p>[6] J. Jiménez, <i>Programación Lógica</i>, Grupo de Lógica Computacional, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Sevilla, España, 2006.</p> <p>[7] R. Kowalski, <i>The Early Years of the Logic Programming</i>, Communications of the ACM, vol. 31, no. 1. Jan., 1988.</p> |
|--|---|