

Grupo CLIP. Dpto. Inteligencia Artificial. Facultad de Informática de la UPM.

## “Estamos aún en la infancia del software, por eso falla”

**Manuel Hermenegildo ha creado el grupo CLIP hace veinte años, con el que intentan desarrollar lenguajes de programación de alto nivel y herramientas para que los programadores puedan crear software sin fallos, eficiente y verificable al mismo tiempo que se va desarrollando.**



Manuel Hermenegildo

**Manuel Hermenegildo obtuvo su doctorado en Ciencias e Ingeniería de la Computación en 1986 por la Universidad de Texas en Austin (EEUU). Ha sido profesor en el Departamento de Informática de dicha universidad y dirigido diversos proyectos y grupos de investigación en la Microelectronics and Computer Technology Corporation en EEUU de 1986 a 1990. En 1990 volvió a España y fundó el grupo CLIP (Grupo de Computación lógica, Lenguajes, Implementación y Paralelismo), que dirige desde entonces en la Facultad de Informática de la Universidad Politécnica de Madrid, donde es Catedrático. También ocupa, desde el año 2003, la Cátedra Príncipe de Asturias en Ciencia y Tecnología Informáticas en los Departamentos de Informática e Ingeniería Eléctrica y de Ordenadores en la Universidad de Nuevo México en EEUU, donde dirige otra parte integrante de este grupo de investigación. Recibió este año el Premio Nacional Julio Rey Pastor en Matemáticas y Tecnologías de La Información y las Comunicaciones.**

### Isabel Gayol Menéndez

“El problema que tenemos con muchos de los objetos cotidianos actualmente es que fallan y casi siempre lo hacen porque en su interior llevan un software. El software, por el momento, falla constantemente”. Así introduce Manuel Hermenegildo, Catedrático de la Facultad de Informática de la UPM, el objeto de estudio del grupo de investigación que dirige: el grupo CLIP o grupo de Computación lógica, Lenguajes, Implementación y Paralelismo.

“Una de las cosas que más nos preocupan es que hoy en día absolutamente todo lo que nos rodea lleva un software incorporado: ascensores, cámaras de fotografía o vídeo, los frenos ABS de los automóviles, los aviones, la cafetera, etc. Al contrario que los aparatos de generaciones anteriores, cuyo funcionamiento se basaba en la mecánica —explica Manuel Hermenegildo— ahora en un alto porcentaje la funcionalidad viene dada por el software, que tiene fallos y que además cambia a un ritmo vertiginoso”.

Los fallos existentes en el software se deben a que los programadores aún no saben cómo desarrollarlo sin errores y de forma que, al mismo tiempo, se ajuste a unas determinadas especificaciones para que luego pueda ser certificado. Es algo en lo que

se está investigando y que aún no está totalmente resuelto. Ello se debe también al hecho de que la informática es una ciencia muy joven, que está realmente en sus inicios. Por eso es necesaria una gran labor investigadora en este campo. La ventaja es que los resultados de dicha investigación pueden tener un enorme impacto económico porque facilitarían un aumento de la productividad en todos los sectores, reducirían sustancialmente los fallos de software y supondrían una ventaja competitiva al poder comercializar a un coste razonable software acompañado de una garantía real.

Todos los artículos a la venta están llenos de sellos y certificaciones y ninguno de los mismos tiene que ver con el software. Las edificaciones, por ejemplo, tienen unas normas de construcción, por lo que cuando un ingeniero diseña un puente, se realizan unos cálculos matemáticos y con ellos se certifica que el puente no se va a caer. Además si le pasara algo a dicho puente se tendrían que asumir las responsabilidades derivadas de su mala construcción. En el caso del software aún no ocurre así. Todavía no disponemos de las técnicas necesarias para que la certificación del software se pueda hacer de forma rutinaria. "En el caso del software estamos como en la época de los maestros masones cuando se construían las catedrales y aún no poseían ni la experiencia ni las matemáticas necesarias para hacer las comprobaciones oportunas. En esa época todo era aproximado y más un arte que una ciencia o ingeniería. La física y posteriormente su aplicación, que es la ingeniería, nos ha dado la tecnología para poder certificar que un edificio no se cae", comenta el catedrático. En este sentido la informática es la física y la ingeniería del mundo virtual en el que se ejecutan los programas de ordenador.

El proceso de escribir software es llevado a cabo por un ingeniero informático

que requiere unas herramientas determinadas para desarrollar su trabajo: el lenguaje de programación, que sería algo similar al idioma en el que se escribe, así como una serie de herramientas en torno a éste. Asimismo dicho lenguaje puede ser de alto o bajo nivel. La diferencia entre ellos es que en un lenguaje de bajo nivel se necesitan muchos pasos intermedios para completar una sola acción, mientras que en un lenguaje de alto nivel se hace de una manera más efectiva y con muchos menos pasos. El objetivo es desarrollar lenguajes de alto nivel que faciliten la labor del programador y consigan que éste sea más productivo y pueda escribir en poco tiempo un programa complejo.

El grupo CLIP tiene muy clara su línea de trabajo y su forma de realizarlo. Su motivación prioritaria es siempre que el fruto de su investigación sea una contribución científica y al mismo tiempo tenga una aplicación práctica. Su regla de oro es que sus descubrimientos sean incorporados a los sistemas que desarrollan. Desde hace 20 años trabajan para que los programadores sean productivos y hagan software complejo, sin fallos y eficiente. Para ello les proporcionan lenguajes de programación de alto nivel y herramientas automatizadas para que al mismo tiempo que escriben un programa puedan localizar sus posibles errores. Estas herramientas permiten también verificar el programa respecto a las especificaciones exigidas, que pueden ser muy variadas: un límite de tiempo determinado para realizar una acción, que sólo se procesen imágenes de cierto tamaño, que el programa no acceda a Internet o a una



Grupo CLIP de la UPM

parte del disco, etc. La principal novedad de su trabajo –además de los lenguajes de programación que desarrollan– está en la forma en la que se combinan todas estas herramientas, además de en la variedad de técnicas de análisis y optimización de programas desarrolladas por el grupo.

“Los programas de ordenador son la única adquisición que en vez de llevar sellos lleva ‘antisellos’. Cuando uno instala un programa lo primero que hace es aceptar que está de acuerdo con que la empresa fabricante del software no se haga cargo de nada si algo no funciona. Lo habitual es justo lo contrario, los sellos de certificación de todos los productos garantizan al consumidor que los fabricantes son los responsables de la calidad y del buen funcionamiento del utensilio que compra. Esta paradoja ocurre por una combinación de varios factores. Por un lado, no se trata de un problema sencillo: los programas de ordenador son hoy en día uno de los objetos más complejos que ha desarrollado el hombre. Por otro, los consumidores de informática aún no son muy conscientes de sus derechos. En lo que a la informática se refiere nos encontramos igual que cuando se vendieron los primeros coches, que no llevaban certificación alguna. Si falla un programa el usuario en primer lugar piensa que ha hecho algo mal, no que es el sistema el que falla. Cuando contactamos con un técnico informático para reparar un ordenador adquirido hace unos meses, es habitual que se nos diga que tenemos que salvar los datos de todos los programas y volver a formatear nosotros mismos el disco duro y reinstalar. Esto sería equivalente a que el técnico de un garaje, además de cobrarnos, nos dijera, seis meses después de comprar un coche, que debíamos desmontar todo el coche y comprar un motor nuevo para volverlo a instalar nosotros mismos. Algo impensable porque los clientes no lo admitirían”.

Además, aunque la gente quisiera quejarse, no hay ningún lugar al que dirigirse porque no existe una Agencia Europea de Calidad del Software o de certificación de software. Pero en realidad hay otro factor más relevante y es que aún no existe la tecnología para poder certificar el software, al menos de forma sencilla.

En algunos casos sí se realiza la certificación oportuna como en aquellos dispositivos en los que entra en juego la seguridad de los ciudadanos como son el software que hace funcionar los distintos componentes instalados en el interior de los aviones o en el freno de los coches. Esto supone un porcentaje muy pequeño del total del software existente. Aún así, la manera de comprobar que dicho software está libre de errores y funciona correctamente es contratando a personal muy especializado, que manualmente realiza cálculos muy complejos. Es un proceso largo y tedioso, para el que muy poca gente en el mundo está cualificada y que, por lo tanto, tiene un coste muy elevado.

### **Líneas de Trabajo**

El equipo de Manuel Hermenegildo pretende automatizar dicho proceso de certificación y que pase a formar parte del entorno de programación para que cualquier ingeniero informático, al mismo tiempo que desarrolla un programa, pueda comprobar las especificaciones o encontrar las razones por las que no se cumplen dichas especificaciones. Se pretende proporcionar un lenguaje de programación del más alto nivel, que permita escribir programas complejos, integrado en un entorno que compruebe simultáneamente si se ajusta a las especificaciones.

El grupo CLIP tiene como línea prioritaria de trabajo plasmar este objetivo mediante el desarrollo de un lenguaje de programación que han creado a lo largo de los años. Se trata de Ciao, un lenguaje de investigación avanzado que utilizan diariamente miles de personas en el mundo, incluido el entorno industrial. En cualquier caso, sus herramientas también se adaptan a otros lenguajes como, por ejemplo, Java.

El desarrollo de un lenguaje de programación va unido siempre al desarrollo de un compilador, que es el encargado de traducir las órdenes de alto nivel a las de bajo nivel. Cuánto más alto sea el nivel del que se parte más complejo resultará el compilador. Por otra parte, normalmente unos programadores se encargan del desarrollo del programa y otros, al final del proceso, de las fases de pruebas y, en su caso, verificación. Por el contrario, el compilador del grupo CLIP verifica y realiza pruebas mientras se desarrolla el programa. Según Manuel Hermenegildo, "No es efectivo verificar al final del proceso porque si no funciona es muy difícil volver a atrás y además supone un importante gasto económico. La novedad que aportamos es que se puede verificar el programa en el proceso de desarrollo del mismo".

El trabajo del equipo da una gran importancia también a la eficiencia: aunque el lenguaje sea de alto nivel debe ser rápido y competitivo. En este sentido, la última gran línea de trabajo del grupo es el *paralelismo*. El paralelismo es necesario para conseguir que las computadoras puedan trabajar más rápido. Los ordenadores que se están fabricando en estos momentos (y que llevan, por ejemplo, la mayor parte de los ordenadores portátiles que se comercializan en la actualidad) disponen ya de dos a cuatro procesadores en su interior, número que, según datos de Intel, se irá duplicando cada uno o dos años aproximadamente. Esto plantea problemas muy importantes ya que los programas existentes están pensados para un procesador único. Por lo tanto un posible camino que toma la investigación informática en este campo es el de paralelizar el programa, es decir, que un compilador sea capaz de ordenar a cada procesador qué parte del programa debe ejecutar, con lo que se conseguiría una mayor efectividad. El compilador busca las partes del programa que son independientes y las reparte entre el número de procesadores del que se disponga. Este proceso es conocido como *paralelización automática* y es otra de las capacidades que tienen los compiladores que desarrolla el grupo.

El grupo de trabajo se denomina CLIP (Grupo de Computación lógica, Lenguajes, Implementación y Paralelismo), y está dividido entre dos continentes por lo que hay un CLIP este y un CLIP oeste. Manuel Hermenegildo vive en la actualidad parte del año en Nuevo México (EE.UU.), aunque tiene su sede en la Universidad Politécnica de Madrid. CLIP está integrado por investigadores de cinco departamentos: el Departamento de Inteligencia Artificial (UPM), el Departamento de Lenguajes y Sistemas Informáticos (UPM), su homónimo de la Universidad Complutense, y los Departamentos de Ciencias de la Computación y de Ingeniería de Ordenadores de la Universidad de Nuevo México (EE.UU.). El personal consiste en doce investigadores de plantilla: 8 profesores y 4 Ramón y Cajal o Juan de la Cierva; 9 doctorandos, de los cuales seis están en Madrid y el resto en Estados Unidos; un técnico de gestión y un técnico informático, de 6 nacionalidades diferentes. Colaboran con grupos de todo el mundo, pero principalmente de Inglaterra, Australia, Dinamarca, Alemania, Francia, Argentina y Brasil.

## Necesidades de la Investigación en Informática

A pesar de que “se han hecho grandes avances en los últimos años en apertura del sistema, mejora de la evaluación y financiación y en los medios destinados a las infraestructuras, aún queda mucho camino que recorrer, ya que en el mundo científico la competencia es feroz. Si se quiere publicar un artículo simplemente se tiene que producir un resultado mejor, y punto. Nadie ampara al que trabaja con menos medios”.

Manuel Hermenegildo afirma que sería necesaria una mayor financiación para los proyectos e infraestructuras y, especialmente, para la remuneración de los investigadores, ya que es la única manera de conseguir atraer a los mejores y que los nacionales no se vayan a ejercer su trabajo a otros países.

Indica también que en España hay una carencia importante: la falta de institutos de investigación en Informática, lo que contrasta con los países de nuestro entorno donde son frecuentes (como los institutos del Max Planck y Fraunhofer en Alemania) e incluso hay macro-organismos a nivel nacional dedicados a la investigación en este área (tales como el INRIA en Francia, un auténtico “CSIC” de a la informática). Para Manuel Hermenegildo “esto supone una carencia en términos de presupuesto y personal cualificado, y una mayor dificultad para crear masa crítica y las sinergias que resultan del trabajo en un centro de este tipo”. En este sentido, destaca el acierto de la próxima creación del IMDEA-Software por parte de la Comunidad de Madrid que espera sea seguido de iniciativas adicionales en otras comunidades y a nivel estatal.

## Proyectos realizados

El grupo ha completado más de 25 proyectos de investigación nacionales e internacionales de los que han resultado publicados más de 150 artículos científicos. Los proyectos activos en la actualidad incluyen el titulado “*Métodos para el Desarrollo de Software Fiable, de Alta Calidad y Seguro*”, financiado por la Comunidad de Madrid y en el que además del grupo CLIP que actúa como coordinador está el grupo BABEL de la Politécnica y los grupos FADOSS y GPD de la UCM. Además trabajan en el proyecto “*Computación móvil verificable y consciente de los recursos*”, MERIT / COMVERS, financiado por el Ministerio de Educación y Ciencia, y en el proyecto “*Movilidad, Ubicuidad y Seguridad*”, MOBIUS, financiado por el Programa Marco y dedicado a cuestiones de seguridad y de movilidad de software, así como en las actividades de la Cátedra Príncipe de Asturias.

## FICHA TÉCNICA

**Centro:** Dpto. Inteligencia Artificial. Grupo CLIP (Computación lógica, Lenguajes, Implementación y Paralelismo). Facultad de Informática. Universidad Politécnica de Madrid.

**Investigador:** Manuel Hermenegildo Salinas

**Dirección:** Campus de Montegancedo. Boadilla del Monte.  
28660 Madrid (Spain)

**Teléfono:** +34-91-336-7435

**Fax:** +34-91-352-4819

**Página web:** <http://www.clip.dia.fi.upm.es/~herme>

**Email:** [herme@fi.upm.es](mailto:herme@fi.upm.es)

**Líneas de investigación:** Análisis, optimización y verificación global de programas. Interpretación abstracta. Entornos avanzados de desarrollo y compilación. Máquinas abstractas. Compiladores paralelizantes. Ejecución distribuida consciente de los recursos. Agentes lógicos. Teoría, diseño e implementación de programación lógica, funcional y por restricciones. Visualización de ejecuciones. Generación automática de documentación con hipertexto y de sitios WWW dinámicos y adaptables. Arquitectura de ordenadores secuenciales y paralelos.