

# Desarrollo de software incorporado en Japón: coevolución de normas en sistemas operativos y redes comunitarias

MASARU YARIME

YASUNORI BABA\*

Entre las industrias de alta tecnología, destaca la del software porque en el desarrollo, la producción y la distribución de sus productos (programas y aplicaciones de cómputo) el capital físico tiene una escasa o nula importancia. Sin embargo, su compatibilidad con la base instalada de equipo (hardware) ha sido determinante como factor de éxito.<sup>1</sup> Hoy en día un creciente número de artefactos y dispositivos —por ejemplo, teléfonos celulares, reproductores de DVD, automóviles y aviones— incorporan programas de cómputo. Las funciones, características y seguridad de estos bienes de consumo dependen cada vez más de la calidad y la disponibilidad del software incorporado<sup>2</sup> en ellos. La complejidad de tales sistemas incorporados está en constante aumento, al igual que la cantidad y la variedad de los respectivos programas.

Ello genera un enorme desafío para el desarrollo del software incorporado, el cual difiere mucho de la elaboración de los sistemas no incorporados, sobre todo en la medida que debe hacer frente a limitaciones específicas, tales como severas restricciones de tiempo; memoria y consumo de potencia limitados; plataformas de cómputo con tecnologías predeterminadas, y costo del equipo.<sup>3</sup> Las tecnologías de de-

\* Centro de Investigación en Ciencia y Tecnología Avanzada, Universidad de Tokio <yarime@zzz.reast.u.tokio.ac.jp> y <baba@zzz.rcast.u-tokio.ac.jp>.

1. David C. Mowery, "The Computer Software Industry", en David C. Mowery y Richard R. Nelson (eds.), *Sources of Industrial Leadership: Studies of Seven Industries*, Cambridge University Press, Reino Unido, 1999.

2. *Embedded software*.

3. Bas Graaf, Marco Lormans y Hans Toeteneel, "Embedded Software Engineering: The State of the Practice", *IEEE Software*, noviembre-diciembre de 2003, pp. 61-69.

sarrollo vigentes no abordan de manera adecuada su efecto específico en el campo del software incorporado; tampoco ofrecen a los desarrolladores las indicaciones necesarias sobre su aplicación en áreas específicas como sistemas automovilísticos, telecomunicaciones y aparatos electrónicos. Más aún, el campo de dichos programas se rige por factores de confiabilidad, factores de costo y tiempos de comercialización, por lo que requiere tecnologías de desarrollo específicamente orientadas.

A la fecha, el poderío japonés en el desarrollo de hardware no ha sido igualado por una posición sólida en lo que a software se refiere, a pesar de los estrechos vínculos entre los desarrolladores de ambos elementos. En el pasado, las políticas nacionales para el desarrollo de capacidades en materia de equipo de cómputo se orientaron a las macrocomputadoras (*mainframes*), y fue justo el éxito de tales políticas lo que en Japón contribuyó a la fragmentación del mercado nacional del software y al predominio de la programación a la medida.<sup>4</sup> Mientras que el entorno más unificado de la arquitectura digital en el mercado estadounidense de la microcomputación contribuyó a un crecimiento espectacular de la industria del software para microcomputadoras en ese país, en el caso de Japón el entorno fragmentado de las normas para hardware retardó el desarrollo de una industria vigorosa de programas y aplicaciones para microcomputadoras.<sup>5</sup> Hoy se podría argumentar, sin embargo, que la creciente importancia del software incorporado puede aumentar la conveniencia y las ventajas competitivas asociadas a la estrategia de vinculación estrecha entre el desarrollo de hardware y de software que han adoptado algunos productores japoneses de sistemas electrónicos.<sup>6</sup>

Estudios recientes sobre innovación destacan la función de las redes de organizaciones para explicar el cambio y la orientación de la evolución tecnológica.<sup>7</sup> Se ha encontrado que los vínculos estrechos entre asociados en redes de colaboración tecnológica fomentan el intercambio de conocimiento, lo que se traduce en un mejor desempeño tecnológico y oportu-

nidades de colaboración entre los socios.<sup>8</sup> Otros estudios realizados sobre redes de innovación explican los beneficios de las relaciones entre organizaciones en términos de difusión de la información y aprendizaje en red.<sup>9</sup> En particular, las industrias caracterizadas por tecnologías complejas como la del software incorporado dependen de grupos de colaboración —comités técnicos, equipos de tarea y cuerpos para la definición de criterios— para decidir sobre el proceso de evolución tecnológica, y los resultados tecnológicos determinan, al tiempo que limitan, la evolución de las redes de cooperación tecnológica entre organizaciones.<sup>10</sup>

El propósito del presente trabajo es investigar la dinámica del desarrollo del software incorporado en Japón. En particular, el interés de los autores se centra en examinar el desarrollo y la utilización en sistemas integrados de diversas clases de sistemas operativos incluidos el ITRON<sup>11</sup> —al que corresponde la mayor participación en el mercado— y otros de uso general como Windows y Linux. En primer lugar se consideran las características de los sistemas operativos específicamente diseñados para sistemas incorporados. Luego de revisar algunos datos sobre la situación actual de la industria del software incorporado en Japón, se hace un recuento detallado de la evolución del proyecto ITRON. Asimismo, se examina el surgimiento de otros sistemas operativos y sus organizaciones asociadas. Al identificar a las personas que han participado en el desarrollo de estos sistemas operativos, se analiza el proceso de innovación en el que han interactuado productores de hardware, vendedores de software, universidades y asociaciones industriales, así como sus comunidades y la coevolución de estándares de los sistemas operativos. Puesto que recientemente se ha observado un auge de los programas de fuente abierta, sobre todo Linux,<sup>12</sup> los autores también han considerado el efecto de este tipo de software en el desarrollo de sistemas incorporados.

4. Yasunori Baba, Shinji Takai y Yuji Mizuta, "The User-driven Evolution of the Japanese Software Industry: The Case of Customized Software for Mainframes", en David C. Mowery (ed.), *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure*, Oxford University Press, Nueva York, 1996.

5. Thomas Cottrell, "Standards and the Arrested Development of Japan's Microcomputer Software Industry", en David C. Mowery (ed.), *op. cit.*

6. David C. Mowery, *op. cit.*

7. Christopher Freeman, "Networks of Innovators: A Synthesis of Research Issues", *Research Policy*, núm. 20, 1991, pp. 499-514, y Walter W. Powell y Stine Gordal, "Networks of Innovators", en Jan Fagerberg, David C. Mowery y Richard R. Nelson (eds.), *Oxford Handbook of Innovation*, Oxford University Press, 2005.

8. Toby E. Stuart, "Network Positions and Propensities to Collaborate: An Investigation of Strategic Alliance Formation in a High-technology Industry", *Administrative Science Quarterly*, núm. 43, 1998, pp. 668-698; Gautam Ahuja, "Collaboration Networks, Structural Holes, and Innovation: A Longitudinal Study", *Administrative Science Quarterly*, vol. 45, 2000, pp. 425-455, y Pek-Hooi Soh y Edward B. Roberts, "Networks of Innovators: A Longitudinal Perspective", *Research Policy*, núm. 32, 2003, pp. 1569-1588.

9. Walter Powell, W. Kenneth W. Koput y Laurel Smith-Doerr, "Inter-organizational Collaboration and the Locus of Innovation: Networks of Learning in Biotechnology", *Administrative Science Quarterly*, núm. 41, 1996, pp. 116-145.

10. L. Rosenkopf y M.L. Tushman, "The Coevolution of Community Networks and Technology: Lessons from the Flight Simulation Industry", *Industrial and Corporate Change*, vol. 7, núm. 2, 1998, pp. 311-346.

11. Industrial Real-time Operating System Nucleus (ITRON).

12. Eric von Hippel y Georg von Krogh, "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science", *Organization Science*, vol. 14, núm. 2, 2003, pp. 209-223, y Georg von Krogh y Eric von Hippel, "Special Issue on Open Source Software Development", *Research Policy*, núm. 32, pp. 1149-1157.

Gracias a los avances en la tecnología de los microprocesadores, la gama de aplicaciones en las que se utilizan sistemas integrados ha aumentado en forma notable. Al principio, tales sistemas se limitaban a aplicaciones industriales como el control de líneas de producción; hoy, registran una acelerada expansión hacia productos electrónicos, de comunicación y, más recientemente, bienes de consumo, incluidos automóviles, sistemas de audio y video, televisores, teléfonos celulares, instrumentos electrónicos, juegos, lavadoras, equipos de aire acondicionado y sistemas de iluminación. El término *sistema incorporado* se aplica ahora a la mayoría de los productos electrónicos que se utilizan en la vida cotidiana. Con el aumento en la gama de aplicaciones de los sistemas integrados sus funciones se han vuelto más complejas; además, la tendencia reciente hacia la digitalización, aunada al incremento en el número de procesos realizados por software en microprocesadores altamente funcionales, ha dado lugar a que estos sistemas tengan cada vez mayor importancia.<sup>13</sup> Puesto que los sistemas integrados de pequeña escala —utilizados sobre todo en artículos de consumo como los teléfonos celulares— se producen por lo general en grandes volúmenes, en comparación con los sistemas integrados de gran escala que suelen encontrarse en los productos industriales, resulta de crucial importancia disminuir sus costos de manufactura. En particular, como resultado de la encarnizada competencia reciente, las compañías han procurado reducir los tiempos de desarrollo de los productos de consumo. Además, sólo en raras ocasiones el software vendido se rediseña, lo que se traduce en un ciclo de vida muy corto para el desarrollo de sistemas.

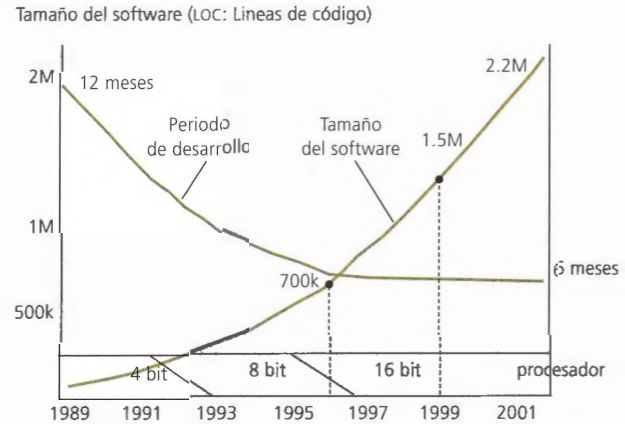
La gráfica 1 presenta un ejemplo del tamaño del software integrado y del periodo de desarrollo para un teléfono celular. Puede observarse que mientras que el periodo de desarrollo de los programas integrados se ha reducido a la mitad —de 12 a seis meses—, el tamaño o la proporción del software ha aumentado más de 10 veces desde finales del decenio de los ochenta, lo que significa un aumento de 200% en la carga de trabajo por unidad.

En la mayoría de los sistemas integrados de pequeña escala, el procesador central, las memorias ROM y RAM, los dispositivos de E/S y algunos otros dispositivos se encuentran todos en un chip llamado microcontrolador (MCU, por sus siglas en inglés) a veces también *microprocesador de un chip*.<sup>14</sup> Puesto que es preciso mantener el costo de desarro-

13. Comité ITRON, "µITRON 4.0 Specification, Ver. 4.00.00", Asociación TRON, 1999.

14. *Ibid.*

TAMAÑO DEL SOFTWARE INCORPORADO EN UN TELÉFONO CELULAR Y PERIODO DE DESARROLLO, 1989-2002 (MEGABYTES)



Fuente: Masayuki Hirayama, *Measures to Improve Quality in the Development of Embedded Systems*, Conferencia sobre Tecnología Incorporada, Yokohama, 20-22 de noviembre de 2002.

llo de los productos finales en el nivel más bajo posible, los recursos de hardware en un microcontrolador —y en particular la memoria— son muy limitados, lo cual entraña un serio desafío para el desarrollo de software. Un microcontrolador muy eficiente tiene distintas clases de procesadores optimizados y diseñados para determinadas aplicaciones. En sistemas integrados de pequeña escala, el aumento en la productividad del software, acompañado de una reducción de sus tiempos de desarrollo, se vuelve un factor importante para el manejo de programas complejos y de gran tamaño.

Para mantenerse a la par de los avances de la tecnología de microprocesadores de alto desempeño es indispensable que los sistemas integrados sean rentables, sobre todo porque hoy en día se les aplica de manera generalizada en más y más artículos de consumo. Asimismo, en la medida que los sistemas integrados se aplican en una amplia gama de áreas, el número de ingenieros de programación que trabaja en sistemas operativos de tiempo real aumenta, y ello hace que su formación sea cada vez más importante. Un estudio reciente señala que los problemas más graves a los que se enfrentan muchos ingenieros que utilizan sistemas operativos de tiempo real en sistemas integrados se refieren a la educación y la estandarización.<sup>15</sup> De acuerdo con los resultados del estudio, no

15. Asociación TRON, *Results of the Survey on the Use of ITRON-Specification OS in the Fiscal Year 1996*, TRON Association, 1997.

son muchos los ingenieros que pueden manejar los sistemas operativos de tiempo real utilizados para los sistemas integrados, y las especificaciones de los distintos sistemas operativos difieren tanto que cambiar de uno a otro implicaría una considerable cantidad de trabajo. También el tamaño y los recursos de los sistemas operativos son excesivamente grandes y la mayoría de sus características y funciones no cumple con requisitos prácticos, lo que origina una serie de dificultades al intentar hacer corresponder un sistema operativo con una aplicación.

El problema más difícil que se encuentra al procurar una especificación de un sistema operativo estandarizada para los sistemas integrados consiste en hallar un equilibrio entre: a) el logro de un mayor desempeño que el hardware permita, y b) el aumento de la productividad del desarrollo de software.<sup>16</sup> En sistemas basados en microcontroladores con estrictas limitaciones en cuanto a recursos de hardware, el máximo desempeño del equipo se logrará sólo si se selecciona con detenimiento un sistema operativo adecuado. Por otra parte, aumentar la productividad del desarrollo de software exige incrementar la abstracción de los servicios del sistema operativo, pero asegurar la portabilidad de los programas —independientemente del hardware en uso— se traduciría en una mayor brecha entre los servicios del sistema operativo y la arquitectura del equipo, brecha que provocaría considerable trabajo indirecto e impediría lograr un óptimo desempeño del hardware. Un compromiso entre ambas metas depende en buena medida del desempeño de los productos incorporados: si bien tendría poco sentido reducir el desempeño en tiempo de ejecución de los sistemas integrados de pequeña escala sólo para mantener bajo el costo del producto final y mejorar su portabilidad, el hecho es que ésta reviste particular relevancia para los sistemas de gran escala, a menudo reciclados.

Los sistemas integrados de pequeña y gran escalas a menudo requieren sistemas operativos de distintas características. Si dependieran de un sistema operativo con muchas características de alto nivel que en realidad no fueran del todo necesarias, los sistemas de pequeña escala verían afectado su desempeño al tiempo que el tamaño de los programas aumentaría. Por otro lado, un sistema operativo con muchas características de alto nivel es útil para los sistemas integrados de gran escala en la medida que permite mejorar la productividad del desarrollo de software. Puesto que los requerimientos para un sistema operativo difieren en función de la escala de cada sistema integrado y de sus características, sería conveniente definir una especificación de sistema operativo

para cada escala de aplicación. Ahora bien, considerando la formación de los ingenieros de programación, la circulación de software y el soporte para herramientas de desarrollo, sería sumamente útil definir una especificación de sistema operativo escalable y adaptable a las necesidades de diversos sistemas integrados. El hecho es que las especificaciones de los sistemas operativos en sistemas integrados se requieren para obtener el máximo desempeño del hardware, contribuir a un aumento en la productividad del desarrollo de software y adaptarse a sistemas de cualquier escala.

#### LA INDUSTRIA DEL SOFTWARE INCORPORADO EN JAPÓN

En comparación con otras industrias con mayor trayectoria, y en virtud de su naturaleza, resulta difícil ubicar la escala de la industria del software incorporado. Con todo, un estudio reciente del Ministerio de Economía, Comercio e Industria de Japón (MECI) aporta información útil sobre la industria del software incorporado en ese país. En el cuadro 1 se presentan datos sobre las industrias del software incorporado, y la industria de servicios de información en Japón.

Con base en estadísticas de las asociaciones industriales de la rama, se calcula que las ventas globales de sistemas incorporados ascienden a 500 000 millones de dólares. De ese total, alrededor de 10% —es decir, 50 000 millones de dólares— se invierte en el desarrollo de sistemas incorporados. Los resultados del estudio del MECI muestran que casi 40% de los costos de dicho desarrollo corresponde a

#### C U A D R O 1

##### INDUSTRIA DEL SOFTWARE INCORPORADO EN JAPÓN

Ventas de sistemas incorporados	500 000 millones de dólares
Costos de desarrollo de los sistemas incorporados	50 000 millones de dólares
Costos de desarrollo del software incorporado	20 000 millones de dólares
Número de compañías relacionadas con los sistemas incorporados	89 000
Número de empleados que trabajan en estas compañías	4.8 millones
Número de ingenieros de programación de software incorporado	150 000
Industria de servicios de información	
Ventas de servicios de información	140 000 millones de dólares
Número de empleados	570 000

Fuente: Comité para el Fomento del Desarrollo de Software Incorporado, Ministerio de Economía, Comercio e Industria, 2004.

16. Comité ITRON, *op. cit.*

programas y aplicaciones, por lo que se calcula que el desarrollo de software incorporado en Japón es del orden de 20 000 millones de dólares.

En Japón hay cerca de 89 000 compañías cuyas actividades empresariales se relacionan con el desarrollo de sistemas incorporados, con un total de 4.8 millones de empleados. De acuerdo con el estudio, conforme aumenta el número de empleados en una compañía, la proporción de ingenieros de programación de software incorporado disminuye. Puesto que los ingenieros relacionados con el desarrollo de dicho software dan cuenta de 3.1% de los empleados en las compañías con más de 5 000 trabajadores, se calcula que en la actualidad su número asciende a 150 000 en Japón. En comparación, la industria de servicios de información produce en conjunto ventas por 140 000 millones de dólares y da empleo a alrededor de 570 000 personas en Japón. Estas cifras sugieren que una parte considerable del sector de la informática corresponde a la industria del software incorporado.

La gráfica 2 muestra las tendencias en la participación de los sistemas operativos utilizados para el software incorporado en Japón.

En Japón se utilizan diversas clases de sistemas operativos para los sistemas incorporados, incluidos ITRON, VxWorks, Windows, Linux y otros sistemas operativos comerciales, así como los de uso específico que algunas empresas han desarrollado de forma interna. Desde mediados del decenio de los noventa la participación de los sistemas operativos basados en las especificaciones ITRON aumentó de manera notable: para 2000 habían alcanzado casi 40% y, si bien su partici-

pación en los últimos años parece estar disminuyendo, hoy día continúan siendo los de más amplio uso en el mercado japonés. Mientras tanto el Linux, que empezó a adoptarse en sistemas incorporados a finales de los noventa, ha aumentado su presencia en el mercado japonés y hoy ocupa ya el segundo lugar, por encima de VxWorks y Windows. A su vez, la participación de sistemas operativos específicos para una empresa ha disminuido de manera constante, lo que denota que un número creciente de compañías prefiere cierta forma de estandarización de los sistemas operativos utilizados para sistemas incorporados; cada vez se favorece más a sistemas abiertos, sobre todo Linux, y en menor grado ITRON.

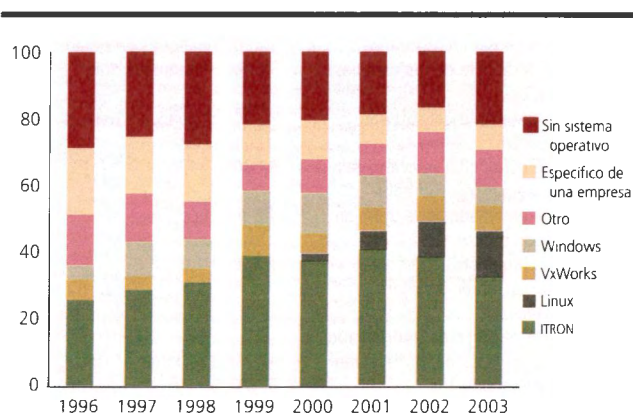
### DESARROLLO DE LAS ESPECIFICACIONES ITRON

El proyecto TRON, que significa “núcleo de sistema operativo de tiempo real” (*The Real-time Operating System Nucleus*), fue iniciado por Ken Sakamura, de la Universidad de Tokio, en 1984.<sup>17</sup> Mediante la colaboración entre la universidad y la industria, el proyecto TRON aspira a producir un concepto totalmente nuevo de arquitectura de cómputo: reconstruir la de los sistemas de cómputo a partir de la visión de una sociedad muy computarizada —la llamada *sociedad cibernética*—, en que la mayoría de los equipos, servicios y herramientas con los que se interactúa día a día cuenten con microcomputadoras interconectadas mediante una red computarizada que les permita funcionar conjuntamente para apoyar las actividades humanas en distintas situaciones.

La política básica del proyecto TRON es que los resultados de su investigación estén disponibles mediante especificaciones abiertas, con base en las cuales cualquiera pueda desarrollar y comercializar libremente sus propios productos. La Asociación TRON se creó como la organización central para desarrollar las especificaciones TRON y también para certificar que un producto se apega a ellas. Las especificaciones TRON definen la interfase de una computadora, no el hardware ni el software con los que opera; determinan también la interfase del sistema operativo, pero no al sistema operativo en sí. El propósito de tales especificaciones es minimizar los costos de desarrollo y facilitar el aprendizaje de usuarios y programadores mediante la compatibilidad de programas y datos. Por consiguiente, la asociación TRON ha adoptado el enfoque de *estandarización holgada*, lo que significa que sólo se define el concepto de diseño: un desarrollador puede instrumentar con libertad un sistema específico siempre que cumpla con el estándar de concepto

G R A F I C A 2

JAPÓN: PARTICIPACIÓN DE LOS SISTEMAS OPERATIVOS UTILIZADOS PARA LA OPERACIÓN DE SOFTWARE INCORPORADO (PORCENTAJE)



Fuente: encuestas de la Asociación TRON, 1997-2004.

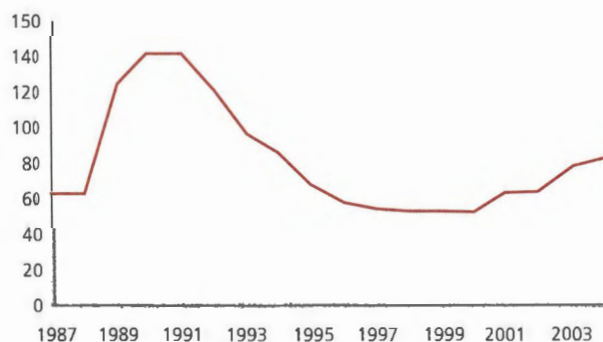
17. Ken Sakamura, “The TRON Project”, *IEEE Micro*, vol. 7, núm. 2, pp. 8-14.

de diseño. El uso de un estándar holgado entraña un equilibrio entre concretar la compatibilidad de los sistemas y permitir la adaptación a nuevas tecnologías.

El proyecto TRON se ha dividido en varios subproyectos básicos y de aplicación: el proyecto ITRON trabaja sobre especificaciones de sistemas operativos de tiempo real para sistemas incorporados y especificaciones relacionadas; el proyecto BTRON, sobre especificaciones de sistemas operativos para computadoras personales y estaciones de trabajo, así como especificaciones relacionadas; el proyecto CTRON, en la especificación de una interfase de sistemas operativos para el control de la comunicación y el procesamiento de la información, y el proyecto TRON HMI, que especifica directrices estándar para una interfase usuario-máquina (*human-machine interface*, HMI) en diversos productos.

La gráfica 3 presenta la tendencia en el número de empresas que pertenecen a la Asociación TRON.

G R A F I C A 3  
NÚMERO DE EMPRESAS QUE FORMAN PARTE DE LA ASOCIACIÓN TRON, 1987-2004



Fuente: encuestas de la Asociación TRON, 1987-2004.

En su etapa inicial, poco más de 60 compañías se unieron al proyecto TRON. Luego el número de empresas participantes aumentó con rapidez, hasta alcanzar 140 a principios del decenio de los noventa, cuando predominaba la percepción de que el proyecto BTRON sería exitoso en virtud de que sus especificaciones en los sistemas operativos se adoptarían en las computadoras personales en Japón. Sin embargo, el sistema operativo BTRON acabó perdiendo la batalla frente a Windows en el mercado de las computadoras personales<sup>18</sup> y

18. Takuma Takahashi y Fujio Namiki, "Three Attempts at 'de-Wintelization': Japan's TRON Project, The US Government's Suits against Wintel, and the Entry of Java and Linux", *Research Policy*, núm. 32, 2003, pp. 1589-1606.

muchas compañías abandonaron el proyecto TRON. A últimas fechas el número ha vuelto a crecer. Una de las razones de este resurgimiento es la adopción generalizada de los sistemas operativos de base ITRON en sistemas incorporados, sobre todo en automóviles y productos electrónicos como teléfonos celulares. El fenómeno se refleja también en el creciente número de compañías que participan en los debates sobre las especificaciones del sistema operativo ITRON. Como se observa en la gráfica 4, luego de un descenso en la etapa inicial, las empresas integrantes del Comité Técnico ITRON han comenzado a aumentar desde mediados de los años noventa.

G R A F I C A 4  
NÚMERO DE EMPRESAS INTEGRANTES DEL COMITÉ TÉCNICO TRON, 1988-2003



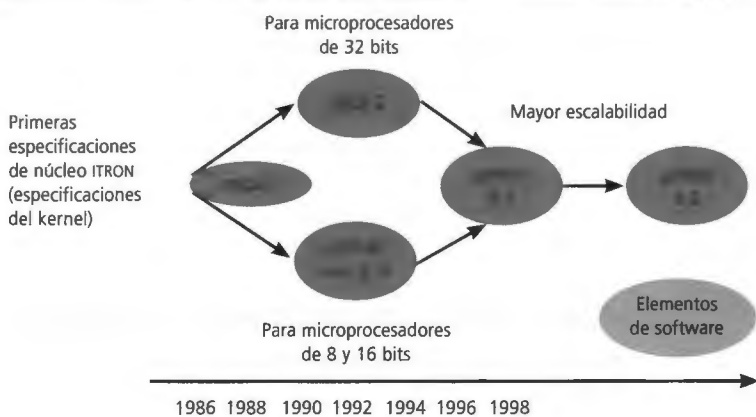
Fuente: encuestas de la Asociación TRON, 1988-2003.

ITRON es una arquitectura para sistemas operativos de tiempo real (RTOS, por sus siglas en inglés) de uso en sistemas incorporados. Desde que el proyecto TRON se inició en 1984, se han formulado y publicado una serie de especificaciones de núcleo o *kernel* de tiempo real ITRON (véase la gráfica 5). El proyecto dio la mayor relevancia a la estandarización de tales especificaciones, puesto que los sistemas de pequeña escala a menudo utilizan sólo funciones de núcleo.<sup>19</sup>

El primer conjunto de especificaciones ITRON, desarrollado en 1987, recibió el nombre de Especificación ITRON 1; desde entonces, muchos núcleos de tiempo real se han desarrollado con base en ellas. En 1989, el proyecto ITRON dio a conocer dos nuevas especificaciones: la especificación  $\mu$ ITRON (ver. 2.0), para sistemas pequeños en microcontro-

19. Comité ITRON, *op. cit.*

EVOLUCIÓN DE LAS ESPECIFICACIONES ITRON



Fuente: Hiroaki Takada, Yukikazu Nakamoto y Kiichiro Tamaru, "The ITRON Project: Overview and Recent Results", Memorias de la Quinta Conferencia Internacional sobre Sistemas de Cómputo y Aplicaciones de Tiempo Real, 1998, pp. 3-10.

ladores (MCU) de 8 ó 16 bits, y la especificación ITRON 2 para sistemas de mayor tamaño en microcontroladores de 32 bits. La especificación  $\mu$ ITRON se ha implantado en gran cantidad de microcontroladores distintos, con memoria y recursos computacionales limitados. De hecho, se le ha aplicado en una amplia variedad de sistemas integrados y ofrece funcionalidad práctica sin grandes requerimientos de memoria, a tal grado que en la mayoría de los principales microcontroladores utilizados en sistemas incorporados se han desarrollado núcleos basados en la especificación  $\mu$ ITRON.

La funcionalidad y el desempeño adquieren especial importancia cuando se pretende aplicar la especificación  $\mu$ ITRON a una amplia gama de campos. Si bien esta especificación no fue originalmente diseñada para microcontroladores de 32 bits, cada vez con mayor frecuencia los núcleos basados en ella operan en tales procesadores. De ahí que en 1993 se pusiera en circulación una edición revisada de la especificación  $\mu$ ITRON 3.0, escalable en microcontroladores desde 8 hasta 32 bits y cuyas funciones de conexión permiten instrumentar un solo sistema integrado en una red.

En la actualidad la Asociación TRON tiene registrados alrededor de 60 productos de núcleo de tiempo real ITRON para más de 40 procesadores de distinta clase. Estos productos incluyen cuatro basados en ITRON 1, 30 en  $\mu$ ITRON 2.0 y ocho en  $\mu$ ITRON 4.0.<sup>20</sup> Aunque algunas empresas estadounidenses de software han llegado a desarrollar núcleos de

especificación  $\mu$ ITRON, la mayoría de los productos corresponde a compañías japonesas, incluidos grandes fabricantes de semiconductores como Toshiba, Fujitsu, Hitachi, NEC y Mitsubishi Electric, junto con empresas especializadas en software como Mispo, eSOL y ACCESS. Puesto que desarrollar un núcleo basado en la especificación  $\mu$ ITRON puede en realidad resultar tarea fácil y breve, muchos usuarios han formulado sus propias versiones para uso interno o casero. También hay muchas otras aplicaciones (que no son productos en sí) y algunas versiones de núcleos  $\mu$ ITRON se distribuyen como software gratuito.

Una de las razones por las que los núcleos  $\mu$ ITRON se han utilizado en muchos campos es que sirven para una gama amplia de aplicaciones: equipos audiovisuales, aparatos domésticos, dispositivos para el manejo de información personal, equipo de oficina, equipos de comunicación, medios de transporte

y dispositivos de control industrial, entre muchos otros. El cuadro 2 presenta algunos de los productos que incorporan núcleos ITRON así como sus fabricantes.

POSIBILIDADES Y LIMITACIONES DE LA NORMALIZACIÓN HOLGADA DE SOFTWARE INCORPORADO

En el diseño de las especificaciones ITRON se adoptaron diversas políticas que podrían resumirse en un concepto: normalización holgada.<sup>21</sup> A efecto de aumentar al máximo el desempeño del equipo y, por consiguiente, incrementar la

C U A D R O 2

PRODUCTOS QUE UTILIZAN SISTEMAS OPERATIVOS ITRON, 2004

Fabricante	Producto aplicado
Toyota	Sistema de control de motor automovilístico Sistema de navegación por voz para DVD
Citizen	Asistente digital personal (PDA)
Yamaha	Instrumento musical electrónico
Ricoh	Fotocopiadora
Techno One	Cámara IP
NEC-Toshiba Space	Satélite

Fuente: Asociación TRON, *Products Based on TRON Specifications*, TRON Association, junio de 2004.

20. Asociación TRON, *Products Based on TRON Specifications*, TRON Association, junio de 2004.

21. Comité ITRON, *op. cit.*

eficiencia en tiempo real, se ha procurado evitar una *virtualización* excesiva del hardware. Para incrementar la adaptabilidad al hardware, y con ello mejorar el desempeño del sistema en su conjunto, se han modificando las especificaciones de los sistemas operativos de tiempo real y se ha ajustado la ejecución interna de acuerdo con las características y el desempeño del equipo. En las especificaciones ITRON, los elementos que se deberán normalizar sin importar la estructura del hardware se distinguen con claridad de los que se pueden optimizar en función del desempeño y las características del equipo. Por otro lado, la estandarización de elementos que reducirían la eficacia del desempeño de ninguna manera es forzosa.

También se toma en cuenta la adaptabilidad a las aplicaciones; es decir, la aproximación que procura mejorar el desempeño del sistema en su conjunto mediante la modificación de las especificaciones del núcleo y los métodos de ejecución interna en respuesta al desempeño funcional del núcleo que las aplicaciones requieren. Puesto que el código objetivo del sistema operativo se crea específicamente para cada aplicación, se considera que el enfoque de la adaptabilidad a las aplicaciones es adecuado en sistemas incorporados. La especificación se diseña de tal manera que cada función del núcleo se mantiene independiente de las restantes y sólo se utilizan las funciones requeridas para cada aplicación. Ello se facilita por el hecho de dar una funcionalidad única a cada sistema. La mayoría de los núcleos de especificación  $\mu$ ITRON se incluye como bibliotecas, y sólo las funciones requeridas se extraen y vinculan con los programas de aplicación.

Se han formulado especificaciones con niveles de soporte definidos para cada cual. A fin de que las aplicaciones se adapten a las diversas clases de hardware, se crearon varias especificaciones con diversos niveles de soporte escalables. Las especificaciones de núcleo de tiempo real incluyen la especificación  $\mu$ ITRON (versión 2.0) para microcontroladores de 8 a 16 bits y la ITRON 2 para procesadores de 32 bits. Con estas especificaciones el usuario puede escalar la funcionalidad conforme resulte necesario e incluir sólo las funciones pertinentes en la ejecución del núcleo. La especificación  $\mu$ ITRON 3.0 separa las llamadas al sistema en diversos niveles de soporte para cubrir procesadores lo mismo de pequeña que de gran escala en una misma especificación.

Compatibilidad y portabilidad no son dos elementos que suelen preocupar a los desarrolladores de software para sistemas incorporados de pequeña escala, ya que es poco probable que los programas se reutilicen en tales sistemas. Por otra parte, normalizar la especificación del núcleo resulta de gran importancia como apoyo en los procesos de educación y capacitación de ingenieros de programación de software, toda vez que homologa conceptos y términos técnicos y, por ende,

facilita la comunicación entre desarrolladores (por ejemplo, posibilita una mayor consistencia en el uso de términos y una nomenclatura para las llamadas al sistema). De esta manera, la normalización holgada es útil para la formación de ingenieros de programación en el proyecto ITRON.

En un principio, el proyecto ITRON se centró en la normalización de especificaciones del núcleo de tiempo real. Sin embargo, conforme los sistemas incorporados se han vuelto más grandes y complejos, la necesidad de normalización de los entornos que circundan al núcleo de tiempo real ha aumentado. En consecuencia, en su segunda fase, iniciada en 1996, el proyecto ITRON comenzó a ampliar su proceso de estandarización para abarcar ya no sólo las especificaciones del núcleo sino también otras relacionadas, sobre todo en componentes de software y herramientas de desarrollo.

En la estandarización de programas y aplicaciones se debe considerar la interfaz para distintos campos, además de las condiciones que propiciarán su desarrollo y distribución. Para preparar tales condiciones, dos cuestiones son importantes. En primer lugar, las diferencias en la ejecución entre núcleos  $\mu$ ITRON impiden asegurar la distribución de elementos de software, problema que se resuelve aumentando la estandarización de núcleos, al tiempo que se mantienen las ventajas de la normalización holgada. En segundo lugar, en la medida que es requisito que muchos programas y aplicaciones tengan capacidades de tiempo real, se requiere un marco que permita la coexistencia de elementos de software que satisfagan sus restricciones de tiempo real y que, idealmente, también permita el uso simultáneo de múltiples componentes.

De noviembre de 1996 a marzo de 1998, un grupo de investigación estudió la funcionalidad necesaria para que un núcleo de tiempo real facilitara la construcción de un sistema de tiempo real riguroso. El comité técnico para la aplicación automovilística de sistemas operativos de tiempo real, establecido en la Asociación TRON, determinó los requisitos para los núcleos de tiempo real en aplicaciones de

## C U A D R O 3

### PRODUCTOS QUE USAN EL SISTEMA OPERATIVO LINUX, 2003

Fabricante	Producto
Sony	Video grabadora Televisión digital Máquina de juego
Sharp	Computadora de bolsillo
Matsushita Electric	Video grabadora

Fuente: Kenichi Nakamura, "Trends in the Embedded Linux Industry 2004", *Software Design*, vol. 2, 2004, pp. 1-6.



control automovilístico, con la idea de incluirlos en la especificación correspondiente. Asimismo, a seis años de entrar en circulación la especificación  $\mu$ ITRON 3.0, la tecnología de los semiconductores ha registrado avances notables, al igual que el desempeño de los procesadores incorporados. También el tamaño de memoria disponible en los procesadores ha aumentado de manera considerable. Por tanto, algunas funciones de núcleo útiles que habían quedado pendientes en la especificación  $\mu$ ITRON 3.0 debido a su costo indirecto, han podido instrumentarse con la tecnología actual.

La creciente necesidad de reconsiderar la especificación de núcleos de tiempo real dio lugar a que se estableciera la cuarta generación de especificaciones ITRON. La especificación  $\mu$ ITRON 4.0 se diseñó sobre todo para mejorar la portabilidad del software y con el propósito de agregar funcionalidad de elementos de software secundarios. En la medida que la complejidad y el tamaño del software incorporado siguen creciendo, aumenta también la necesidad de que las aplicaciones puedan cambiar con facilidad a núcleos distintos. Así, la portabilidad del software desarrollado en un núcleo ITRON se ha vuelto un asunto importante en términos de la distribución de los elementos de software.

En suma, el concepto común de normalización holgada del ITRON significa que algunas partes de una especificación que reducirían el desempeño del equipo no tienen por qué ser estandarizadas forzosamente, y su instrumentación —ya sea en hardware o en aplicaciones— queda en manos del programador del software. Como tal, este concepto ha dado más importancia a la adaptabilidad de equipos y procesadores, y está orientado a lograr el máximo desempeño de diversas plataformas de equipo. Por otra parte, en la medida en que la política de normalización holgada permite distintas versiones de códigos fuente, con frecuencia podría ocurrir que una clase de software que funcionó sin problemas en cierto sistema operativo con base ITRON, fallara en otro. Esta deficiente portabilidad entraña un grave obstáculo, toda vez que la cantidad acumulada de *middleware* y de programas de aplicación reviste crucial importancia.<sup>22</sup>

## EVOLUCIÓN DE DIVERSOS SISTEMAS OPERATIVOS Y REDES COMUNITARIAS

A partir del reconocimiento de estos problemas, el proyecto TRON creó el Foro T-Engine en junio de 2002, con la participación de 22 empresas. El número de participantes ha aumentado considerablemente desde entonces, y para

septiembre de 2004 sumaba 452 empresas y organizaciones, lo mismo internacionales que nacionales. Los 30 miembros del consejo incluyen usuarios de sistemas integrados como el principal operador de telefonía móvil de Japón, NTT DoCoMo, y el proveedor de autos Denso, además de grandes fabricantes de chips.<sup>23</sup>

Desde el inicio del proyecto TRON en 1984, algunos de sus conceptos iniciales han cambiado. El caso más notorio es el de la estandarización holgada, concepto que finalmente se abandonó porque limitaba la creación de entornos de desarrollo poderosos e impedía el transporte fácil de *middleware* de un procesador a otro. Asimismo, el código de fuente T-Kernel, sistema operativo del T-Engine, sólo está disponible para los integrantes del Foro T-Engine en los términos de la licencia T-Kernel. Si bien las empresas miembro están autorizadas a modificarlo, es preciso que notifiquen al Foro lo que hacen, de manera que pueda mantenerse la compatibilidad, y por tanto la portabilidad, del software. A diferencia de lo que ocurre con la licencia pública general (LPG), los desarrolladores no están obligados a renunciar a sus derechos de confidencialidad por las porciones modificadas que agreguen; asimismo, la distribución binaria por sí misma también es posible. Puesto que la responsabilidad de la emisión recae en el Foro T-Engine, la titularidad de la propiedad intelectual queda clara, lo que no ocurre con códigos de programa de una comunidad abierta con muchos contribuyentes no identificados.<sup>24</sup>

En otra dirección del amplio espectro del proyecto ITRON, en noviembre de 2000 el doctor Hiroaki Takada, profesor en la Universidad de Nagoya, comenzó el proyecto TOPPERS, acrónimo de Toyohashi Open Platform for Embedded and Real Time Systems (Plataforma Abierta Toyohashi para Sistemas Integrados y de Tiempo Real).<sup>25</sup> Takada fue alumno doctoral del doctor Sakamura en la Universidad de Tokio y participó activamente en el establecimiento de las especificaciones ITRON como uno de los principales miembros del proyecto. A sabiendas de que la excesiva diversidad de los sistemas operativos con base ITRON genera el problema de una disminuida portabilidad del software e inversiones duplicadas, el proyecto TOPPERS se propuso instrumentar sistemas operativos de tiempo real con códigos de fuente abierta, más allá de la mera definición de especificaciones inicialmente considerada en el proyecto ITRON. En lugar de centrarse en sistemas operativos de uso general, como el

23. Foro T-Engine, *Forum Guide*, 2004.

24. Ken Sakamura y Noboru Koshizuka, "T-Engine: The Open, Real-time Embedded-systems Platform", *IEEE Micro*, vol. 22, núm. 6, 2002, pp. 48-57.

25. Hiroaki Takada, *Current State of the TOPPERS Project*, Conferencia TOPPERS, Tokio, 3 de junio de 2004.

22. *Middleware*: soporte lógico personalizado; software que actúa como traductor universal entre tecnologías y protocolos. [N. de la t.]

Linux, la meta del proyecto TOPPERS es desarrollar sistemas operativos diseñados de manera específica para diversos sistemas incorporados aprovechando la experiencia acumulada en las actividades del proyecto ITRON. Hoy en día la iniciativa cuenta con 60 integrantes entre organizaciones educativas y empresas, incluida Toyota, la mayor compañía automovilística de Japón. Dado que la industria automovilística ha sido identificada como uno de los principales campos de aplicación, el proyecto recientemente puso en circulación un sistema operativo de tiempo real basado en la especificación OSEK/VDX, el cual es de facto el estándar adoptado en forma generalizada por fabricantes y proveedores de autopartes europeos.<sup>26</sup>

En sistemas incorporados en los que la capacidad de tiempo real riguroso es crucial, sin importar la presión que el sistema experimente, debe haber una respuesta a un evento en un tiempo predefinido. Para los sistemas operativos de uso general, como Linux o Windows, el manejo de tareas en tiempo real riguroso no resulta fácil. Por lo general pueden responder a eventos en un tiempo predeterminado, pero ése no siempre es el caso, lo que se conoce técnicamente como tiempo real laxo. Linux experimenta este fenómeno por diversos motivos, sobre todo porque no puede adelantarse a ciertas operaciones del núcleo.<sup>27</sup>

Con todo, como el número de industrias que utilizan Linux, sobre todo en servidores PC, no cesa de aumentar en todo el mundo, muchas compañías han empezado a utilizar sistemas operativos en sistemas incorporados. Se considera que el principal motivo del creciente interés en Linux estriba en que es un sistema operativo de fuente abierta, lo que significa que cualquiera está en libertad de desarrollar y poner en circulación productos que lo utilizan, sin restricciones. Sony en videograbadoras, televisión digital y máquinas de juego; Sharp en PBA y Matsushita Electric en videograbadoras son algunos de los principales usuarios del sistema operativo Linux.

Los desarrolladores de sistemas incorporados y los fabricantes de productos comerciales, sin embargo, se enfrentan al hecho de que la condición de fuente abierta también

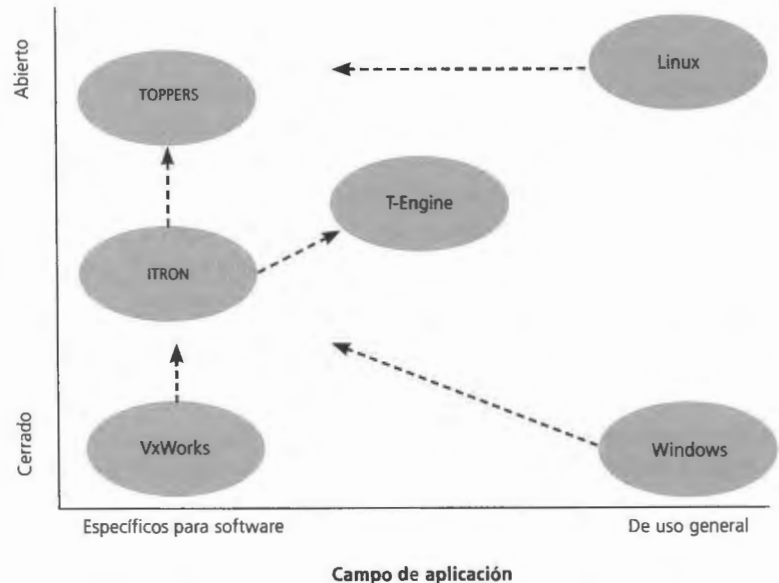
presenta problemas de mantenimiento, soporte y control de calidad. Para atender estas cuestiones en noviembre de 2000 se creó el Consorcio para el Linux Incorporado de Japón (Japan Embedded Linux Consortium, Emblix), con la participación de 24 empresas y organizaciones. Sus miembros son hoy día más de 100 compañías y organizaciones.<sup>28</sup>

Por su parte, el VxWorks de Wind River ha ganado una sólida reputación como sistema operativo para sistemas incorporados.<sup>29</sup> Desde que empezó a circular en 1987 ha sido utilizado ampliamente para sistemas de control de gran escala en los campos aeronáutico y militar, sobre todo en Estados Unidos.<sup>30</sup> De hecho, VxWorks es el sistema operativo que se usó en los robots exploradores Spirit y Opportunity, contruidos por el Laboratorio de Propulsión a Chorro, que hace no mucho aterrizaron en Marte.

La gráfica 6 ubica los sistemas operativos ITRON, T-Engine, TOPPERS, Linux, VxWorks y Windows en términos

**G R A F I C A 6**  
**SISTEMAS OPERATIVOS PARA SOFTWARE INCORPORADO**

**Código fuente**



Fuente: elaboración del autor.

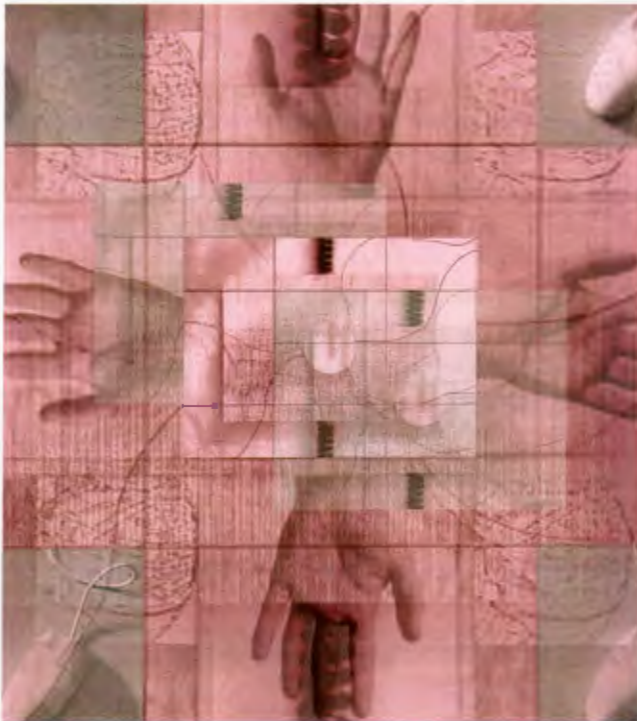
26. Proyecto TOPPERS, *Development of the Open-source OS Based on the OSEK Specification*, boletín de prensa del Proyecto TOPPERS, 26 de abril de 2004.

27. Armand Marchesin, "Using Linux for Real-time Applications", *IEEE Software*, septiembre-octubre de 2004, pp. 18-20.

28. Emblix, *Development Environment Standard Specifications (Rev. 1)*, Japan Embedded Linux Consortium, 14 de agosto de 2002.

29. Gary Nutt, *Operating Systems*, Pearson Education, Boston, 2004.

30. Kunio Inokari, "Current State of Real-time OS", *Interface*, núm. 8, 2004, pp. 36-39.



de la apertura del código de fuente y de sus campos de aplicación.

En lo que respecta a campos de aplicación, Windows y Linux —originalmente diseñados como sistemas operativos de uso general— han comenzado a ampliar su alcance, a efecto de incluir sistemas incorporados. Por su parte, en un principio pensado para usarse en sistemas incorporados, el proyecto ITRON está evolucionando en dos direcciones: el T-Engine tiene el propósito de incluir sistemas más generales, en tanto que el proyecto TOPPERS se mantiene en el campo de los sistemas integrados, como ocurre con el VxWorks. En términos de apertura del código de fuente, Linux y Windows representan los dos extremos; el proyecto ITRON podría ubicarse en algún punto entre ambos polos, ya que sólo establece especificaciones del sistema operativo, pero su instrumentación depende de las empresas, lo que da lugar a distintas versiones de códigos fuente. Se observa una tendencia hacia sistemas operativos de fuente abierta; incluso Windows comienza a orientarse en ese sentido desde que Microsoft empezó a revelar a fabricantes y desarrolladores —para su modificación— el código fuente de Windows CE, la versión específica para sistemas incorporados.<sup>31</sup> Este panorama cambiante en el que coexisten sistemas operativos, programas y aplicaciones de códigos

31. Microsoft, *Windows CE Shared Source Premium Licensing Program*, boletín de prensa, Microsoft, 10 de abril de 2003.

abiertos y cerrados dará a las empresas un amplio alcance para estrategias híbridas.<sup>32</sup>

Con el propósito de identificar las interacciones del establecimiento de estos sistemas operativos y con la gente que ha intervenido de manera activa en su diseño, se realiza un análisis de la red de los participantes en las comunidades de software incorporado, incluidos el proyecto ITRON, el Foro T-Engine, el proyecto TOPPERS y Emblix. Se incluyó también a otro grupo relacionado con el software incorporado en Japón: el grupo SESSAME (Society of Embedded Software Skill Acquisition for Managers and Engineers). Creado en 2000 por el profesor Yoshinori Iizuka, de la Universidad de Tokio, con el propósito de educar y brindar la información necesaria a ingenieros de programación de software incorporado,<sup>33</sup> dicho grupo cuenta en la actualidad con más de 200 miembros, provenientes de empresas fabricantes de equipo y desarrolladoras de software, así como de organizaciones educativas.

Para obtener información sobre las personas que trabajaron en el establecimiento de los sistemas operativos, se revisaron las listas de integrantes de las organizaciones que apoyan los sistemas operativos, así como los grupos de trabajo pertinentes al interior de tales organizaciones.<sup>34</sup> La red bimodal incluye tanto a los grupos de desarrollo como a las personas que en ellos participan.

Se identificó a 294 personas que participaron activamente en el proyecto ITRON; 55 en el proyecto T-Engine; 20 en el proyecto TOPPERS; 55 en Emblix y 209 en el grupo SESSAME. Puesto que algunas de las personas participan en múltiples grupos, el total de 586 participantes es menor que la suma simple de los integrantes de cada grupo.

La persona que más activamente ha trabajado en el desarrollo de software integrado en Japón es Hiroaki Takada. Al tiempo que se desempeñaba como líder en el proyecto TOPPERS, participaba en tres comunidades de sistemas operativos: ITRON, Linux y SESSAME. Su papel central en éstas se confirma al medir los grados de centralidad e intermediación (*betweenness*) que le corresponden, en ambos casos los más altos entre todos los participantes. Le siguen cinco personas que han estado trabajando en tres grupos: Ken Sakamura, Hiroshi Monden, Kiichiro Tamaru, Hideo

32. Joel West, "How Open Is Open Enough? Melding Propriety and Open Source Platform Strategies", *Research Policy*, vol. 32, 2003, pp. 1259-1285.

33. Yoshinori Iizuka, *Invitation to SESSAME*, SESSAME, 24 de marzo de 2001.

34. Emblix, *Development...*, *op. cit.*; "Specification of Hybrid Configuration between Linux and RTOS (Rev. 1)", Japan Embedded Linux Consortium, 10 de agosto de 2002; TRONWARE, T-Engine, Personal Media, Tokio, 2003 y 2004; SESSAME, *SESSAME Participating Members*, 2004; Proyecto TOPPERS, *TOPPERS Project: Introduction and Invitation*, julio de 2004; Association TRON, *The TRON Project: 20<sup>th</sup> Anniversary*, TRON Association, 2 de junio de 2004.

Inokari e Hiroyuki Kato. En particular, el doctor Sakamura ha estado dirigiendo el proyecto ITRON durante los últimos 20 años y recientemente lo amplió para organizar el Foro T-Engine; por su parte, los señores Monden, de NEC, y Tamaru, de Toshiba, ambos directores de Emblix, se desempeñaron recientemente como presidente y vicepresidente, respectivamente, del Comité para Fomentar el Desarrollo de Software Incorporado organizado por el MECI. Puesto que todas las personas arriba mencionadas en algún momento trabajaron juntas en el proyecto ITRON, podría inferirse que hay algunos canales de conexión entre las comunidades de sistemas operativos.


Para investigar con mayor profundidad los vínculos entre comunidades, se generó otra gráfica unimodal a partir de la original, en este caso conservando los cinco grupos. La gráfica 7 ilustra los resultados. Los números cercanos a los nodos representan el total de participantes en los grupos, en tanto que los próximos a las líneas denotan el número de personas que participan en los dos grupos conectados, según se refleja en el grosor de las líneas.

Se observaron algunos traslapes en los participantes entre los proyectos ITRON y T-Engine o TOPPERS. Esto era de esperarse, toda vez que los proyectos T-Engine y TOPPERS se basan ambos en las especificaciones ITRON. Lo que resulta interesante señalar es que son relativamente más los elementos en común entre ITRON y Emblix, cuyos sistemas

operativos difieren entre sí. Esto sugeriría que el proyecto ITRON, que comenzó hace 20 años, ha logrado de manera eficaz aportar experiencias valiosas a los participantes, quienes han podido luego aprovecharlas en el desarrollo de sistemas operativos para sistemas incorporados con características distintas.

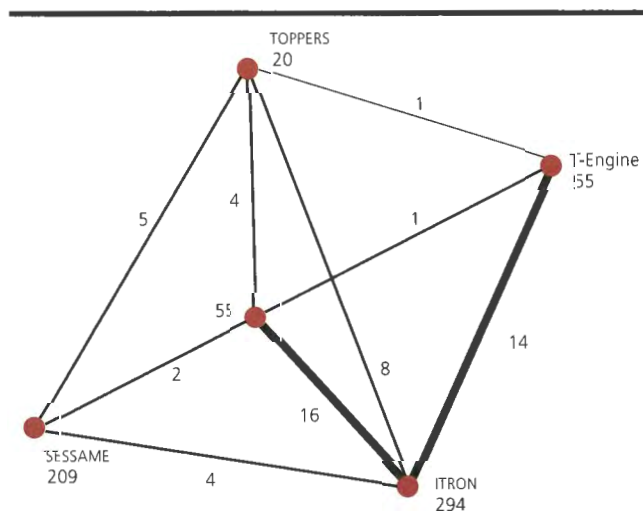
## CONCLUSIONES

En este trabajo se examina la evolución de los sistemas operativos para sistemas incorporados en Japón. Mediante el análisis de la estructura traslapada de las organizaciones de sistemas operativos y sus participantes fue posible detectar la coevolución de las normas para sistemas operativos y las comunidades de desarrollo de software incorporado. La existencia de redes de individuos en dicha industria podría ser de utilidad en un momento en que la cooperación y el establecimiento de alianzas entre las comunidades de sistemas operativos están adquiriendo relevancia. Al usar sistemas operativos comunes, las compañías podrían reducir el creciente costo de desarrollar un sistema operativo a partir de cero, y centrar sus esfuerzos en el desarrollo de programas de aplicación.

Esto se ha demostrado en forma bastante clara en la industria automovilística. En la medida en que el software incorporado empleado en los sistemas de control es cada vez mayor y más complejo, los fabricantes de automóviles y autopartes han empezado a hacer causa común con organizaciones de desarrollo de programas y aplicaciones en pro de la estandarización del software. En 2002 se creó en Europa la organización Autosar (Automotive Open System Architecture), con debates entre sus principales integrantes, entre ellos BMW, Bosch, Volkswagen y DaimlerChrysler, para desarrollar y establecer un estándar abierto en la industria para la arquitectura electrónica automovilística.<sup>35</sup> Más recientemente, Toyota y Nissan establecieron en Japón una organización similar, JASPAR, con el propósito de estandarizar el software y las redes para sistemas electrónicos de control para vehículos.<sup>36</sup> Por otra parte, no se observa un movimiento similar en Estados Unidos, donde al parecer cada fabricante de automóviles mantiene su propio sistema con los proveedores de autopartes. Sería interesante investigar con mayor profundidad las condiciones, las tendencias y las formas de cooperación y competencia en materia de sistemas operativos en las diferentes industrias y países. 

### G R Á F I C A 7

RED UNIMODAL DE LAS COMUNIDADES DE SOFTWARE INCORPORADO EN JAPÓN



Fuente: elaboración propia con información de Emblix, Proyecto TOPPERS y Asociación ITRON.

35. Autostar, *Background*, 2004.

36. Toyota Motor Corporation, 2004, *Nissan and Toyota Join Efforts to Standardize Software and Networks for Electronic Vehicle Control Systems*, boletín de prensa de Toyota Motor Corporation, 4 de septiembre de 2004.