



¿Cómo interactúa el hardware con el software?

Este artículo profundiza en el funcionamiento de las máquinas computadoras e intenta elucidar la cuestión de cómo o en que sentido puede interactuar el software con el hardware. Para efectos de facilitar la exposición, se concibe una máquina sumadora ad-hoc sobre la cual se realiza el análisis detallado de tal interacción, llegando finalmente a mostrar como se traduce dicha interacción en el formalismo de las máquinas de Turing y los computadores digitales.

Por Jorge Maturana Palma, 25 de Abril de 2009.

Introducción: la metáfora del computador

El vertiginoso desarrollo experimentado en las últimas décadas por las tecnologías de la información y su exitosa radicalización en casi todas las áreas o actividades del quehacer social, organizacional, económico, y en la interacción del hombre con su medio y recursos, ha puesto muy de moda, no sólo entre los científicos sino que también en la mayoría de las personas, una popular analogía entre la actividad mental-cerebral humana y los computadores electrónicos.

Por ejemplo, a la luz de este símil, un dualista cartesiano no dudaría en afirmar que el hardware, es decir, los componentes mecánicos, eléctricos y electrónicos son el cuerpo o sustancia material del sistema, y que el software o conjunto de programas computacionales, sería el sustrato no material o la mente inmaterial que habita en la máquina. El hardware patente a la percepción sensorial, no explicaría por sí sólo el comportamiento exhibido por el sistema, ya que por más que se escarbe entre los circuitos eléctricos, componentes mecánicos y flujos electromagnéticos de la máquina, nada habría que se asemejara ni siquiera medianamente a un razonamiento, pensamiento o inferencia. Debe haber en la máquina algún otro tipo de sustancia que sea portadora de tales propiedades y que sea causante directa de los "estados mentales" constatados en la máquina. Consecuentemente, el software inmaterial depositado en el hardware sería el espíritu que anima la máquina y articula sus componentes mecánicos inyectándole de esta forma vida y movimiento. De aquí que el sistema propiamente tal, no sería otra cosa que el software circunstancialmente residente en un hardware particular. Finalmente, para el dualista, el funcionamiento del sistema completo sólo podría explicarse en términos de algún tipo de interacción entre el hardware o sustancia material y el software o la sustancia inmaterial que determina su comportamiento, y el problema radicaría en hallar los términos precisos en que se produce la interacción entre tales componentes.

Por otro lado, desde una perspectiva de la psicología computacional, la metáfora se traduciría de la siguiente manera:

i. La arquitectura básica de los computadores se ajusta a un modelo que, en sus aspectos estructurales, se compone de los siguientes elementos:

- Un cerebro compuesto básicamente de una unidad de control y una
- unidad de pensamiento
- Una memoria de corto plazo
- Una memoria de largo plazo
- Organos aferentes de entrada
- Organos eferentes de salida

ii. La comunicación entre estas partes se efectúa a través de una red nerviosa de buses o canales, que permiten transportar las señales eléctricas de un lugar a otro. El control y la coordinación de todos estos componentes es realizada por una unidad de control, que forma parte del "cerebro" del conjunto. Dentro de la máquina circulan las corrientes eléctricas rigurosamente sincronizadas gracias a un reloj interno que envía impulsos con una frecuencia determinada estableciendo el ciclo base de la unidad de control. Por otra parte, la Unidad de Pensamiento, es la zona del "cerebro" encargada de efectuar las operaciones

aritméticas y lógicas, tales como el manejo de los operadores "and", "or", "not" e inferencias del tipo "if (condición) then..." lo que permite canalizar una determinada secuencia de pasos y bifurcarla hacia otras secuencias dependiendo si se verificó o no la condición.

iii. La memoria de corto plazo, ubicada en una zona del "cerebro", está dividida en compartimientos o celdas cada una de las cuales posee una dirección y puede contener distintos tipos de información. Esta memoria es de muy rápido acceso pero de poca capacidad, por lo cual, la escritura de nueva información se realiza con el consiguiente borrado de parte de la que existía anteriormente. La memoria de corto plazo es esencialmente volátil ya que su contenido se pierde cuando se produce una desconexión temporal o se escriben nuevos contenidos muy extensos. No obstante, en esta memoria siempre se encuentra presente la información necesaria para realizar la actividad actual o más reciente.

iv. El propósito de la memoria de largo plazo es conservar grandes cantidades de información y por una cantidad de tiempo indefinida. A diferencia de la memoria a corto plazo, su acceso es más lento pero su capacidad es sustancialmente mayor, y su contenido posee una mayor persistencia en el tiempo.

v. A través de los órganos de entrada, la máquina recoge la información del mundo exterior y mediante los órganos de salida, entrega las respuestas al mundo exterior. Asociados a estos órganos existe un mecanismo altamente especializado en codificación y decodificación de datos que permite convertir las señales del mundo externo en las señales de naturaleza electromagnéticas reconocidas por la máquina y viceversa. En este mecanismo puede localizarse la memoria sensorial, puesto que aquí se encuentran presentes los estímulos provocados por las percepciones sensoriales más recientes.

La arquitectura descrita corresponde a la que clásicamente se conoce como el modelo de Von Neumann, del cual ofrecemos una adaptación en la siguiente figura:

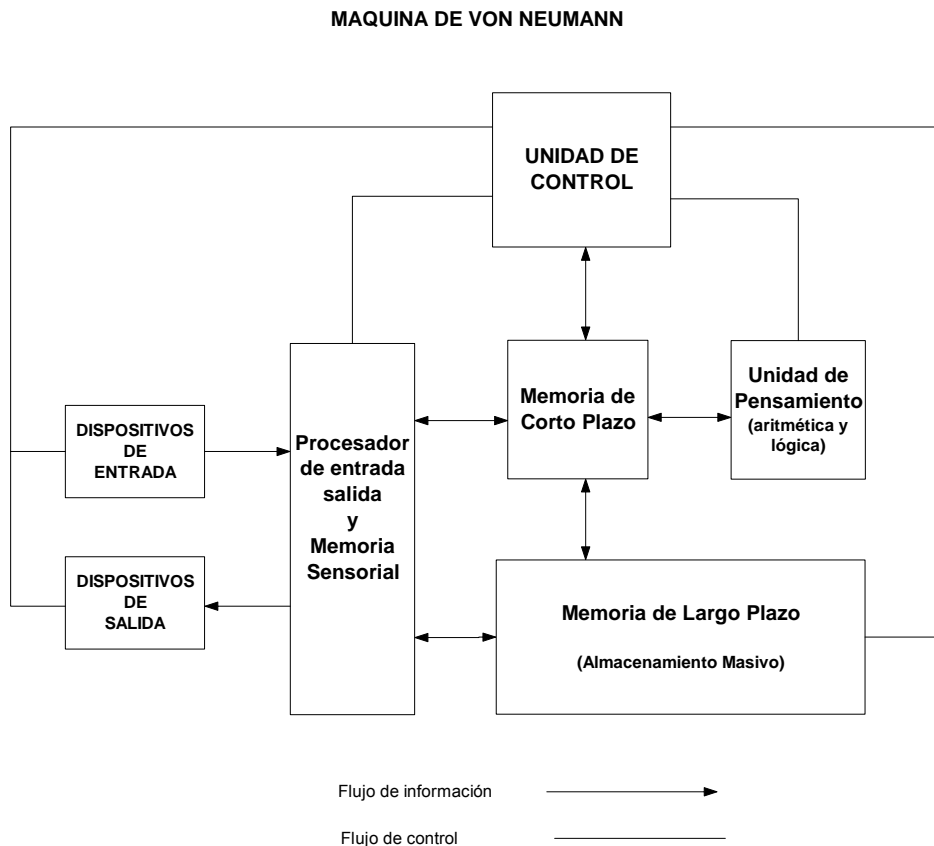


Figura 1.- Modelo clásico de un Computador Digital de propósito general

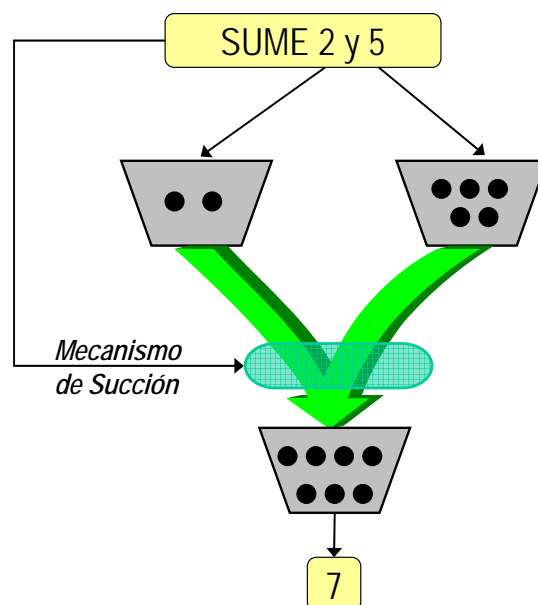
La interacción del hardware con el software

En relación a esta atrayente y poderosa metáfora, desde una perspectiva técnica sabemos efectivamente que un sistema de computación se compone de programas computacionales que operan en el hardware, y también sabemos que el programa es independiente del hardware en el sentido que puede ejecutarse en cualquier computador digital de similares características.

Sin embargo, para apreciar claramente bajo qué términos se verifica algún tipo de interacción entre el hardware y el software, dejando de lado las complicaciones de los circuitos electrónicos que podrían inducir a confusión, recurriremos al diseño de una máquina sumadora que no obstante su simpleza, nos proporciona una conveniente analogía para estos propósitos.

La máquina en cuestión obedece a una instrucción simple del tipo "SUME a y b", donde la palabra "SUME" representa la acción a realizar y las variables a, b corresponden a números enteros no negativos sobre los cuales se realizará la operación, en este caso, de suma de enteros. La arquitectura básica de la máquina sumadora consta de dos casillas de entrada donde se almacenan las variables a sumar, y una casilla de salida donde se registra el resultado después de la acción de suma.

Las casillas de entrada están conectadas mediante ductos tubulares al vacío que convergen simultáneamente a la casilla de salida; en la intersección de los tubos se ubica una bomba de succión que al operar succiona las bolillas puestas en las casillas de entrada en dirección a la casilla de salida. Los números se representan mediante bolillas puestas en los respectivos casilleros, de tal forma que la cardinalidad del conjunto de bolillas en cada casillero corresponde a los valores de los números de entrada y salida, respectivamente. Cada vez que la máquina recibe una instrucción, por ejemplo "SUME 2 y 5", primero realiza la representación física de los literales numéricos mediante el poblamiento de bolillas en las casillas de entrada, y después ejecuta el verbo de acción, que en nuestro caso, consiste en la activación del mecanismo de succión que transporta las bolillas de los dos casilleros de entrada al casillero de salida; posteriormente se realiza la traducción de la cantidad de bolillas en la casilla de salida al respectivo literal numérico que las representa. La siguiente figura muestra el modelo de esta máquina y la operatoria de suma recién descrita.



Ahora, si sobre la base de esta máquina alguien nos preguntara cómo es que interactúa el hardware (máquina compuesta de casilleros, bolillas, ductos, bomba de succión, etc.) con el software (conjunto de sentencias o instrucciones de la forma "SUME a y b"), la respuesta sería de la siguiente forma:

- a) Existe una función de representación que traduce los literales numéricos asociados al verbo de acción de la sentencia imperativa, en determinadas cantidades de bolillas físicas que son puestas en las casillas de entrada.
- b) Para el verbo de acción de dicha sentencia existe una función de transformación que lo traduce en el accionamiento de una bomba de succión que transporta las bolillas desde ambos casilleros de entrada al casillero de salida.
- c) Finalmente se ejecuta la inversa de la función de representación que traduce las bolillas en el casillero de salida al respectivo literal numérico.

En otras palabras, lo esencial de la interacción entre software y hardware, es un par de funciones que traducen sucesos lingüísticos en sucesos físicos, en el caso de los literales numéricos se trata de una función de representación, y en el caso del verbo de acción, se trata de una función de proyección que relaciona el verbo con la activación física de un mecanismo de succión (que por lo demás, nada tendría que ver con el contenido semántico de dicho verbo). Por otro lado, nótese que si reemplazamos bolillas por núcleos magnéticos, ductos por buses de comunicación y bomba de succión por circuitos electrónicos, tenemos una ajustada analogía con un computador digital sin perder en absoluto sustancialidad conceptual. El secreto de la interacción estaría dado entonces por la adecuada construcción de funciones que conecten dominios lingüísticos (inmateriales), con dominios físicos (materiales).

Esta misma característica se encuentra presente en el formalismo de las máquinas de Turing, donde estas funciones se encuentran representadas por la función de transición directa, definida como la proyección:

$$\delta: Q \times T \rightarrow Q \times T \times \{L, N, D\}$$

donde,

Q es un conjunto de estados finito y no vacío de la unidad de control

T es un alfabeto de cinta

$d \in \{L, N, D\}$ representa la dirección en que se mueve el cabezal de la cinta, según:

si $d = L$, la cabeza se mueve un cuadrado a la izquierda

si $d = N$, el cabezal no se mueve

si $d = D$, el cabezal se mueve un cuadrado a la derecha

De esta forma, el funcionamiento de una máquina de Turing queda espontáneamente determinado por la definición que se establezca para la función de transición directa:

$$(q_1, A_1) \rightarrow (q_2, A_2, d) = \delta(q_1, A_1)$$

lo que significa que (1) la unidad de control se encuentra actualmente en el estado q_1 y el símbolo A_1 se lee del cuadrado actual mediante el cabezal, y (2) al realizar un movimiento atómico, la unidad de control de la máquina cambia al estado q_2 , el símbolo A_1 se reemplaza por A_2 , y la cabeza de la cinta se mueve un cuadrado en dirección d . Lo destacado de esta función, es que relaciona símbolos de estado y alfabéticos, con movimientos físicos.

Volviendo al ejercicio de nuestra máquina sumadora, si ahora se insistiera en la pregunta de si el software es independiente del hardware, entonces deberemos diferenciar entre dos acepciones diferentes de la sentencia "SUME a y b". Por un lado, esta

sentencia posee un contenido semántico que denota la abstracción matemática de sumar dos números (SUME* a y b), y por el otro, también significa la activación y realización del mecanismo de succión (SUME** a y b).

Evidentemente, la connotación matemática de "SUME* a y b" constituye una abstracción que es independiente de cualquier implementación operativa. Sin embargo, la sentencia "SUME** a y b" sólo tiene sentido como parte de una función específica que refiere correlacionalmente a un dominio material. Si el software es lo segundo, entonces no es independiente de la implementación física o material donde se proyecta, aunque si posiblemente, en algún sentido, del tipo de material con que se construya (las bolillas podrían ser de madera o metal, o en vez de bolillas podríamos utilizar moscas, pero la acción de succionamiento no la podemos soslayar). No obstante, en la práctica, no nos presta mucha utilidad hablar de succiones de bolillas, preferimos hablar de sumas de enteros, pero este modo de denotar constituye una interpretación ajena al sistema.

Por otro lado, si se tuviera un mapa acabado de los circuitos e interrelaciones electromagnéticas de un computador, manipulando directamente las magnetizaciones binarias de los circuitos del hardware estaríamos indirectamente modificando el programa y produciendo un cambio de comportamiento en el sistema. Análogamente si se dispusiera de un mapa completo del programa, mediante su manipulación produciríamos modificaciones en la estados magnéticos de la máquina y alterando la respuesta del sistema. La razón de tal coincidencia es simple: ambos aspectos obedecen a una misma función de transición que correlaciona dominios conceptuales y físicos. La "metáfora del computador" no sabemos si es un buen ejemplo de algún aspecto del funcionamiento del cerebro, pero si es un claro paradigma de cómo puede lograrse coincidencia estructural, en este caso sintáctica, de una configuración material y un algoritmo lógico (no-material).

Conclusiones

Finalmente, nos queda pendiente un aspecto que nos interesa destacar: afirmamos que el software mantiene dependencia directa con la funcionalidad material específica del hardware (SUM** es sinónimo de succión y no otra cosa), y que en cierto sentido sería independiente del tipo de materiales que compongan el hardware ¿pero, hasta qué punto?: una específica estructura material si tiene que ver con el tipo de material con que esta se construya, puesto que no toda sustancia posee la plasticidad y propiedades necesarias para configurar eficazmente una determinada estructura y funcionalidad material.

Jorge Maturana Palma es Socio Principal de la consultora en psicometría Orgánica Ltda., Ejecutivo de Tecnologías de Información de un importante Retail, Socio fundador de la Asociación de Informáticos UTE-USACH, Director para Chile de la Asociación Latinoamericana de Testing de Software (ALATS). De profesión Ingeniero Informático, posttitulado en Gestión Informática, es Magister en Filosofía de las Ciencias, Diplomado en Neuropsicología y Magister(c) en Psicología Social Aplicada

Cómo citar este documento:

Maturana, J. (2009):¿Cómo interactúa el hardware con el software?. Extraído el dd-mmm-aaa de http://www.jorgematurana.cl/dv/P09/pct/PCT0906a_Como_interactuan_HW_y_SW.pdf