

CONCEPCION MULTINIVELICA Y CUASIDESCOMPONIBLE DE SISTEMAS COMPLEJOS, APLICACION A LA INFORMATICA,

por

D. Lampaya Latorre

F. Sáez Vacas

Laboratorio de Ordenadores, Ciber
nética y Teoría de Sistemas.

E.T.S.I. Telecomunicación. Madrid

RESUMEN

Como contribución al desarrollo y comprensión de sistemas complejos se construyen unas conceptualizaciones no ambiguas de las nociones de "complejidad" y "sistema complejo". A partir de ellas se muestran los cuatro factores generadores de complejidad en un sistema: el observador, el entorno, los componentes y las interacciones. En base a estos dos últimos se propone una clasificación de los sistemas según la complejidad de su estructura, reseñando la emergencia de propiedades, y justificando la concepción multinivélica-cuasidescomponible de sistemas complejos. Por último, se comenta brevísimamente la aplicación de estas ideas a la Informática en aspectos tales como las metodologías de desarrollo, la programación estructurada, las reglas de visibilidad, protocolos, sistemas operativos y un modelo de sistemas de tratamiento de la información.

1. INTRODUCCION.

El interés por el estudio (diseño análisis, descripción) de sistemas complejos se debe a factores muy diversos, que van desde el desarrollo de las ciencias sociales y naturales hasta los logros espectaculares de la tecnología.

Trabajar con esta clase de sistemas presenta dificultades propias que se ven acentuadas por:

- . la heterogeneidad en cuanto al sentido de los conceptos "complejidad" y "sistema complejo".
- . la falta de una visión que permita aplicar principios generales para enfrentarse a la complejidad
- . cierta subestimación de herramientas que por su difícil formalización o gran generalidad pueden parecer poco potentes.

La informática es un campo en que la necesidad del estudio de la complejidad es creciente, no sólo

por la instrumentación de sofisticadas arquitecturas hardware y software, sino también por la variedad, riqueza y constante evolución de las técnicas que la componen y que requieren en su conjunto un estudio como sistema complejo.

Partiendo de estas consideraciones, se presenta aquí una visión global de los conceptos "sistema complejo" y "complejidad"-en una acepción más amplia e infrecuente que la de complejidad algorítmica profundizando en el aspecto estructural que, entre otros, justifica la concepción multinivélica y cuasidescomponible de sistemas complejos. En síntesis brevísima de un estudio actualmente en curso, se muestra la utilidad de la aplicación de estas ideas a la Informática. La comunicación se cierra con un capítulo de conclusiones.

2. SISTEMAS COMPLEJOS

2.1. sistema

La Figura 1 expresa una abstracción del enfoque sistémico. El observador (O) centra su atención sobre un objeto en el universo, a partir de los que define un sistema (S) y el entorno (E) en el que existe. Entre S y E pueden intercambiarse entradas (e) y salidas (s).

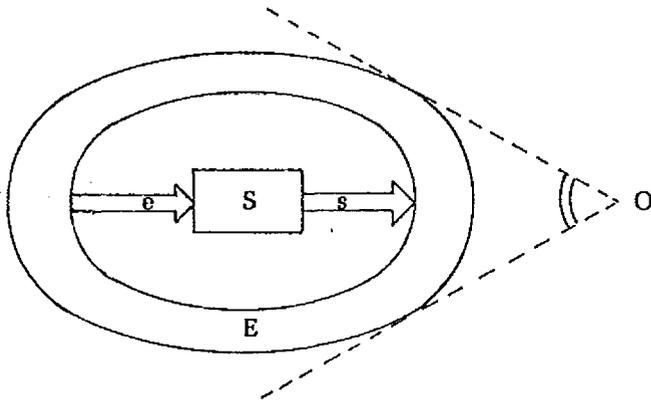


FIGURA 1. Enfoque Sistémico

El grafismo invita a la descripción funcional o comportamental $S; s = S(e)$.

Asimismo, es posible la visión estructural del sistema, en la que se toman en cuenta un conjunto de componentes (C) de S y unas interacciones (I) a un determinado nivel de abstracción. S es, entonces, IUC.

2.2. Sistemas complejos. Complejidad.

Un sistema complejo es "aquél" constituido por un gran número de partes que interactúan en una forma no sencilla... Dadas las propiedades de las partes y las leyes de interacción, no es un problema trivial inferir las propiedades del todo". [1].

Aunque esta definición traslada el problema de la conceptualización de lo complejo a la de la interacción no sencilla, se aprecia en ella que las partes (componentes) y las interacciones van a influir en que un sistema sea percibido como complejo. A estos dos factores deben añadirse el entorno, del que dependen algunas de las interacciones, y el observador, que definió los tres aspectos anteriores: entorno, interac-

ciones y componentes.

La complejidad es la magnitud que mide en qué grado un sistema es complejo. Aunque no existe uniformidad de criterios sobre cómo debe establecerse esta magnitud, resulta razonable generalizar la idea de que "la complejidad de una cadena binaria es la cantidad mínima de información para definir la cadena" [2], admitiendo entonces, que la complejidad de un sistema es la cantidad mínima de información necesaria para caracterizarlo.

Con este "metro", se aprecia intuitivamente que, al añadir componentes y/o interacciones a un sistema, se incrementa su complejidad al precisarse mayor información para definirlo.

3. ESTRUCTURA

Aunque observador y entorno son factores fundamentales en la complejidad de un sistema, se deja su desarrollo para otra ocasión, clasificándose a continuación los sistemas de acuerdo únicamente a su complejidad estructural.

3.1. Tipos de interacciones

Si se concibe la interacción como la solicitud de un servicio, existen tres posibilidades entre dos componentes de un sistema:

- $\square \square$: no hay interacción entre ellos, son independientes.

- $\square \rightarrow \square$: un componente es el maestro que solicita el servicio del esclavo. El esclavo puede informar de la conclusión del servicio.

- $\square \leftrightarrow \square$: los componentes se solicitan servicios mutuamente; permutan dinámicamente el papel maestro-esclavo.

3.2. Tipos de estructuras

Dados (n+1) componentes, la complejidad estructural del sistema crecerá al hacerlo el número de interacciones. Partiendo del tipo más sencillo, resulta:

- Sistema No-interactivo: Fig. 2a

- Sistema n-interactivo: Figura 2b. Hay una definición rígida del papel maestro-esclavo. Un componente es esclavo, a lo sumo, de otro, aunque puede ser a su vez maestro de un número cualquiera de componentes. Resulta en la je-

arquía clásica.

- Sistema nivel-interactivo: Figura 2c. Los componentes se agrupan por niveles, de tal forma que los miembros del nivel n pueden ser maestros de todos los del nivel $(n+1)$ y esclavos de todos los del $(n-1)$. Cada nivel es un estrato.

- Sistema fuertemente-interactivo Figura 2d. Cada componente puede solicitar los servicios de todos los demás, y viceversa.

Un sistema no suele ser representante puro de un tipo, sino una mezcla de varios.

- mayor posibilidad de introducir redundancia, potenciando la capacidad de contrarrestar procesos de degradación.
- modularidad.
- rapidez de actuación,

Sin embargo, se pierden otros factores, tales como la simplicidad, que hacían el análisis o la síntesis más inmediatos. (Paradoja no sorprendente, ya que todo sistema comporta propiedades favorables y desfavorables para algún criterio de evaluación). ¿Existe algún compromiso entre ambas situaciones?. La respuesta puede ser el diseño multinivélico y cuasidescomponible de sistemas complejos.

4. MULTINIVELES- |CUASI|DESCOMPO-NIBILIDAD.

4.1. Visión multinivélica

El sistema se analiza o sintetiza como un conjunto de niveles, de tal forma que los componentes de un estrato pueden solicitar sus servicios a todos aquellos del estrato inmediatamente inferior (si existe) y a los de su propio estrato (si existen) pero no a los del resto de los estratos. Se tiene un híbrido de sistema nivel-interactivo, por las relaciones entre estratos, y fuertemente-interactivo, por las relaciones intra-estratos.

La visión multinivélica, sin llegar al grado máximo de complejidad estructural que sería el sistema fuertemente-interactivo, permite ya disponer, en zonas localizadas del sistema, de fuertes interacciones bidireccionales. Estas zonas se unen al resto por interacciones unidireccionales, más simples debido a la distinción estática maestro-esclavo.

4.2. |Cuasi|descomponibilidad.

4.2.1. Descomponibilidad.

Un sistema es descomponible, a un nivel de abstracción dado, si a dicho nivel de abstracción es NO-interactivo. Su análisis o síntesis se reduce al de cada uno de sus componentes a ese nivel.

4.2.2. Cuasidescomponibilidad.

Introduciendo la fuerza de las interacciones -aspecto hasta ahora soslayado a pesar de su importancia [3]- puede definirse un umbral de tal forma que, despreciando las

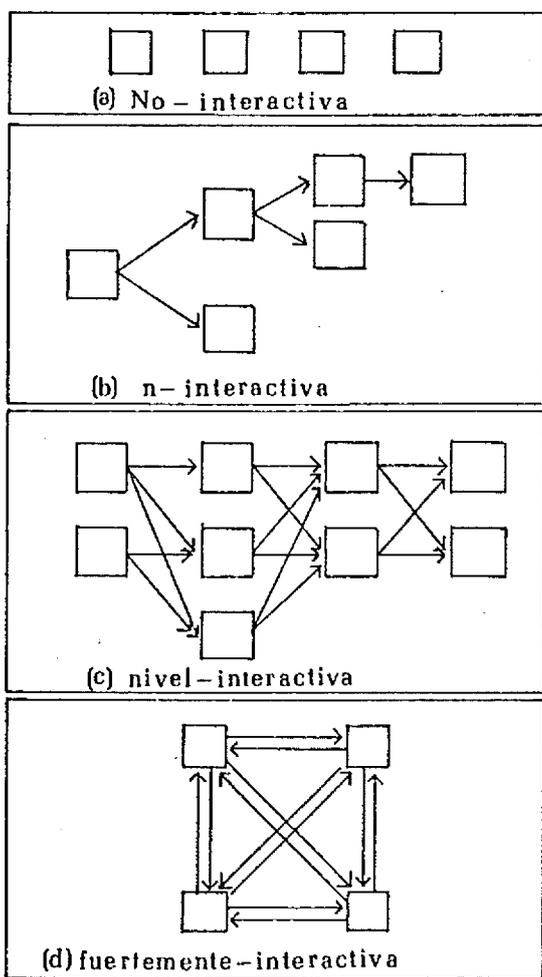


FIGURA 2. Tipos de Estructuras

3.3. Propiedades

Al crecer la complejidad de la estructura aparecen rasgos tales como:

- diálogo entre componentes. (Servicio bidireccional).
- emergencia de propiedades, al ser "el todo distinto de la suma de las partes."

interacciones de fuerza inferior a dicho valor umbral, el sistema se simplifique al aparecer como descomponible. Iterando el proceso pueden llegar a aislarse componentes lo suficientemente simples como para que sean manejables. Un sistema que presenta esta gradación de interacciones se denomina cuasidescomponible.

En ciertas circunstancias, el hecho de desprestigiar interacciones introduce errores. Para evitarlo, cuando sea necesario, habrá que simularlas. La simulación será tanto más simple cuanto más débiles sean las interacciones por las que se descompone el sistema.

4.3. Conjunción

Conjugando multiniveles y cuasidescomponibilidad se puede llegar a analizar o sintetizar sistemas complejos en base a un ataque ordenado de la complejidad, bajo control del observador.

La estructura multinivel aconseja un límite al tipo de interacciones; la cuasidescomponibilidad, a su fuerza. La primera nos guía en la construcción de complejidad; la segunda, en su análisis. Sin embargo el carácter circular de ambos procesos, construcción-análisis, provoca la confusión de sus límites.

Se concluye que la concepción, desde el análisis o la síntesis, de los sistemas complejos, debe guiarse por una visión multinivélica-cuasidescomponible.

5. APLICACIÓN A LA INFORMÁTICA.

La informática produce o se aplica cada día a sistemas de creciente complejidad. Para poder afrontar con eficacia su diseño, implementación y mantenimiento aparecen progresivamente nuevas herramientas, herramientas que responden, cada vez en mayor grado, a un planteamiento multinivélico-cuasidescomponible de los sistemas informáticos. En apretadísima síntesis pueden citarse, entre otras:

5.1. Metodologías de desarrollo

Insisten en el esquema top-down* bottom-up de diseño y realización. Es la muestra más pura y directa de la conjunción sistema multinivélico-cuasidescomponibilidad.

Son numerosos los estudios que demuestran las ventajas de costo, fiabilidad, modificabilidad y mantenibilidad de los sistemas realizados según este planteamiento.

5.2. Programación estructurada

La programación estructurada, con sus tres operadores concatenación selección e iteración -necesarios y suficientes para la construcción de cualquier algoritmo-, supuso una revolución en su momento. Sin embargo, no hizo sino establecer fuertes limitaciones estructurales para posibilitar la construcción de programas complejos en forma eficaz. Estas fuertes limitaciones estructurales radican en el hecho de que un programa estructurado es un árbol (n-interactivo). La permisión del GOTO en ciertas situaciones (aproximadamente las "exception" de ADA) y la inclusión de concurrencia -por ejemplo, operador COBEGIN, COEND- es un paso hacia estructuras más sofisticadas, incorporando sus propiedades (cfr. 3.3), pero sin perder nunca el carácter multinivélico y cuasidescomponible de los sistemas resultantes.

5.3. Reglas de Visibilidad.

Las reglas de visibilidad tienen por objeto favorecer la definición de la arquitectura del programa resultante como n-interactiva o nivel-interactiva. Los componentes se declaran al nivel mínimo necesario para que sólo los componentes de nivel superior que van a solicitar sus servicios puedan hacerlo. Se elimina la posibilidad de interacciones no necesarias.

Al nivel más externo sólo se aprecian los rasgos globales del sistema, mostrando los niveles más internos características peculiares o específicas. Además, la declaración de un objeto es la única información que sobre su uso tiene el maestro que solicite sus servicios. (Visión funcional del componente).

En esta línea, acostumbrados (?) ya a lenguajes tipo PASCAL, empieza a ser mayoritariamente reconocida la utilización de tipos de datos abstractos. En ellos, el tipo de datos se caracteriza por las operaciones que sobre él pueden realizarse, ignorando su im-

plantación en memoria y la gestión de dichas operaciones.

La influencia de estas y otras herramientas se han reflejado en numerosas técnicas informáticas que se han orientado a concebir sistemas con base más o menos sistemática en estos principios. Entre ellas se consignan algunas de las más generales o difundidas:

5.4. Protocolos

La arquitectura multicapa definida por la ISO [4] para permitir la realización de sistemas abiertos.

5.5. Sistemas Operativos.

siguiendo el ejemplo clásico del P.H.E. [5] cada vez son más los sistemas operativos jerarquizados (terminología de Dijkstra) y que, de acuerdo con lo dicho previamente, se denominarían multinivélicos.

Por último, la propia riqueza de ramas, herramientas y otros elementos que conforman la informática parecen pedir que se la estudie como un sistema complejo en sí misma. En tal sentido puede considerarse el siguiente modelo de principio de los Sistemas de Tratamiento de la Información (STI).

5.6. Sistemas de Tratamiento de la Información.

Se define un STI como un conjunto de ordenadores y terminales conectados por una red. Un STI se caracteriza por dos niveles fundamentales de descripción (Fig. 3):

- El nivel I (Interacciones), que resalta las propiedades emergentes debidas a las interacciones entre los componentes de utilidad general del nivel G. Según su función puede tratarse de un PROTOCOLO, un SISTEMA OPERATIVO, una BASE DE DATOS, la APLICACION.

- El nivel G (Generalidad), que describe los componentes generales empleados para realizar las funciones del nivel I. Corresponden a dos grandes familias -HARDWARE, SOFTWARE- a las que se añadirá antes o después el FIRMWARE.

Los componentes de I se materializan, tras el paso por una serie de niveles de refinamiento, en componentes de G.

En [6] se detalla el modelo. Se ha presentado aquí para reflejar la simplicidad con que un enfoque multinivélico-cuasidescomponible permite describir los aspectos de un STI, a pesar de su gran complejidad, a un nivel de abstracción determinado.

6. CONCLUSIONES

En una presentación extremadamente escueta de un tema tan amplio como el que es objeto de esta comunicación, se debe optar entre un planteamiento global o su aplicación a un ejemplo concreto.

Se ha preferido el primer camino, realizando una presentación introductoria, que servirá así de marco de referencia para futuras ocasiones, al mismo tiempo que pueda acaso suscitar interés en otras personas ante el generoso campo que el enfoque ofrece.

La profundización sobre las métricas de la complejidad; la repercusión que en la misma tienen sus cuatro productores: observador, entorno, componentes e interacciones; la caracterización de la fuerza de las interacciones son, entre otros, puntos de la Teoría de Sistemas sobre los que se sigue investigando. Lo mismo puede decirse sobre la aplicación de la concepción multinivélica-cuasidescomponible a la Informática, no sólo en los aspectos mencionados por ejemplo, el modelo STI- sino también en otros como bases de datos, arquitecturas hardware, etc.

Las páginas precedentes han descrito un hilo conductor. Los sistemas complejos tienen propiedades que los hacen muy interesantes. En la medida en que se encuentren técnicas para analizar y sintetizar la complejidad, será posible una utilización más eficaz de tales sistemas, y su descripción y estudio, más sencillos y económicos. Actualmente parece que la herramienta más potente para trabajar con sistemas complejos es su concepción multinivélica-cuasidescomponible, que está relacionada con redundancias (constricciones), normalización, abstracción y otros factores.

La Informática -generadora de sistemas complejos- emplea ya este enfoque, a veces de modo inconsciente. Es preciso generalizarlo,

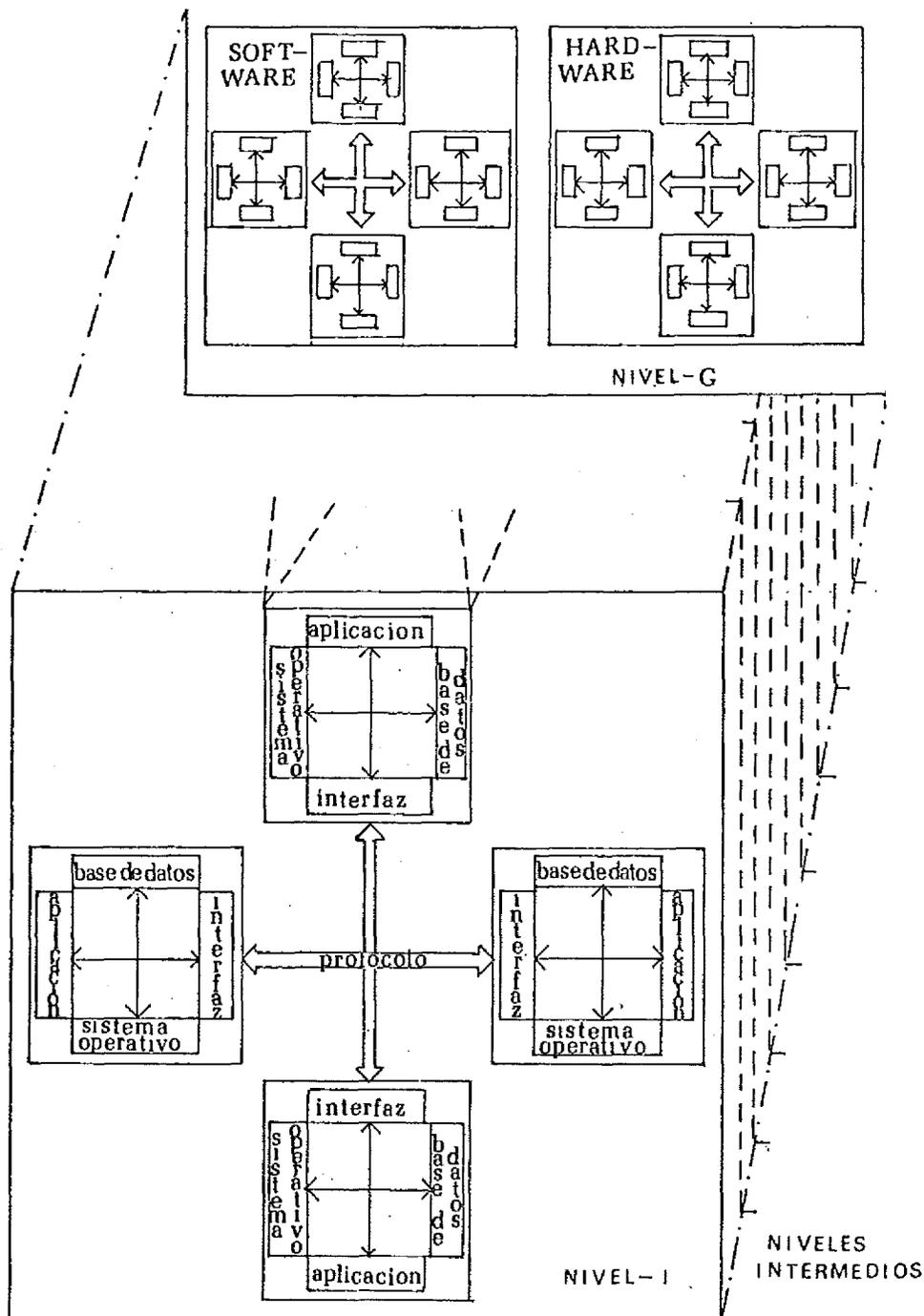


FIGURA 3. Modelo de los STI.

explicitarlo y difundirlo.

7. BIBLIOGRAFIA:

- (1) SIMON, H.A., "The Architecture of Complexity" en "The Sciences of the Artificial", The M.I.T. Press, Cambridge, Mass pp. 84-118; 6th pr, 1978.
- (2) CHAITIN, G.J., "Information - Theoretic Computational Complexity", IEEE T. Inf. Theory - IT-20, n°1, pp. 10-15, Jan. 1974
- (3) CONANT, R.C., "Detecting Subsystems of a Complex System", IEEE T. Sys. Man Cyb. SMC-2, pp. 550-553, Sept. 1972.
- (4) JARDINS, R. des, WHITE, G., "ANSI Reference Model for Distri-

buted Systems", Proc. Computer Communications Networks, pp. 144-149, IEEE Comcon - Fall 78; Sept. 1978.

- (5) DIJKSTRA, E.W., "The Structure of the "THE"-Multiprogramming System", Comm. of the ACM, V.11, pp. 341-347, May 1968

LAMPAYA, D., "System Complexity Views as a Tool for Simplifying Computer Science Views" Lab. Ordenadores, Cibernética y Teoría de Sistemas, E.T.S.I Telecomunicación, UPM/ETSIT/LOCTS/02-81, Madrid, Febrero 1981.