

# APLICACIÓN DE LA TEORÍA GENERAL DE SISTEMAS AL ANÁLISIS DE LOS SISTEMAS INFORMÁTICOS

Yuriy D.Zubenko<sup>1,2</sup>, Zulma Cataldi<sup>2</sup>, Fernando J. Lage<sup>2</sup>

1. Universidad Argentina J. Kennedy

2. Facultad de Ingeniería - Universidad de Buenos Aires. informat@mara.fi.uba.ar

## Resumen

Utilizando la metodología del análisis de sistemas y la teoría general de sistemas se pueden resolver tres tareas pertinentes a los sistemas informáticos en estadios diferentes: el análisis de sistema de computación (en forma estática), el análisis de los sistemas de programación (en estática y dinámica) y el pronóstico de la evolución de los sistemas de computación (evolución cíclica). Para ello, se usa el esquema o modelo general del sistema informático para representar cada tarea del sistema en forma de modelos matemáticos sencillos tales como vectores, matrices, expresiones lógicas, ecuaciones, etc.

Estos resultados obtenidos han sido publicados libros y en artículos, los que confirman las posibilidades del análisis de sistemas. [1]

## Introducción

La representación del análisis de sistemas como nueva ciencia de los sistemas exige las aplicaciones prácticas de nuevos conceptos. En este artículo se presenta la resolución de tres tareas con ayuda del análisis de sistemas y la teoría general de sistemas: el análisis de sistemas de computación, el análisis de sistemas de programación y el pronóstico de los sistemas de computación.

Las propiedades informáticas son algunas de las diferentes propiedades de la materia (químicas, físicas, económicas, biológicas, informáticas, etc.). Las propiedades se pueden organizar de diferentes modos diferentes. Cuando las propiedades organizadas realizan alguna función (F), se puede decir, que esas propiedades están organizadas en forma de un sistema. Recordando la definición de sistema:

“Un sistema (S) es un conjunto de propiedades del objeto real (N) que realizan una sola función” (F) [1].

Todos los sistemas representan una jerarquía de teorías correspondientes. La cima de esa jerarquía es el sistema general abstracto que esta siendo estudiando por la teoría general de sistemas. [1]

Los sistemas informáticos ocupan su lugar entre otros tipos de sistemas y tienen subtipos: sistemas de computación, sistemas de ayuda, sistemas operativos, sistemas de compilación, sistemas de programación, sistemas de inteligencia artificial, etc. De acuerdo con la definición de sistema nosotros se puede representar un sistema informático tal como se ve en el esquema de la figura 1.

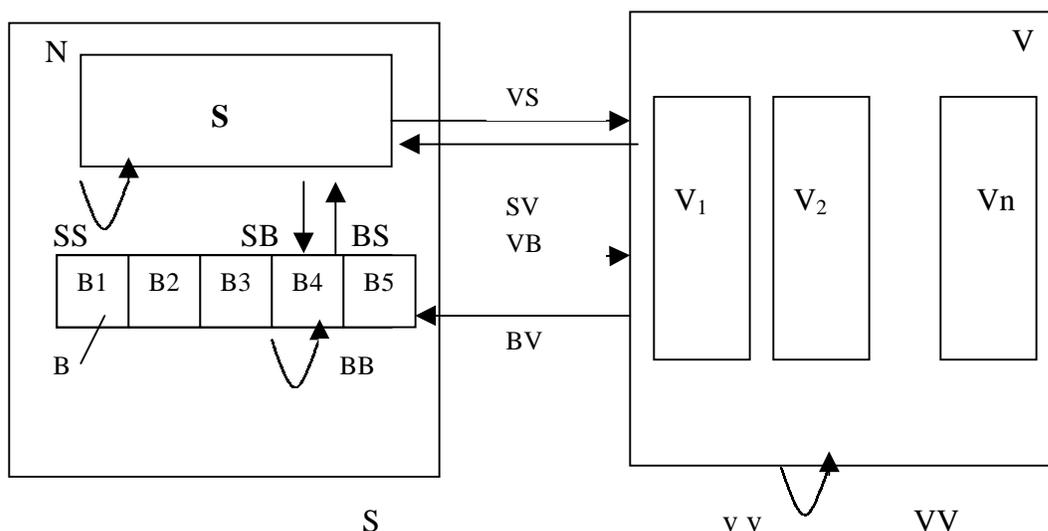


Figura 1. Esquema de un sistema informático. Claves: N: objeto real; V: medio exterior; S: sistema; B: base del sistema; B<sub>1</sub>: personal (calificación); B<sub>2</sub>: personal (ergonómico); B<sub>3</sub>: hardware; B<sub>4</sub>: software; B<sub>5</sub>: base de datos; V<sub>1</sub>, V<sub>2</sub>, ..., V<sub>n</sub>: partes del medio exterior; SS, SB, BS, SV, VS, BB, BV, VB, VV: relaciones.

Este esquema general se puede aplicar a los sistemas concretos para resolver las tres tareas antes citadas: análisis del sistema de computación, análisis del sistema de programación y pronóstico de evolución de los sistemas de computación

### El Análisis del sistema de computación [2]

Un sistema de computación se puede representar por su función F:

$$I_S = F(I_E) \dots (1),$$

Donde  $I_E$ : información de entrada,  $I_S$ : información de salida.  $I_E \sim V_1 S$ ,  $I_S \sim S V_1$ ,  $V_1 \in V$ ,  $V_1$ : objeto de gestión o usuario del sistema,  $SS$ : autogestión, puede ser alguna parte más de  $I_E$  y  $I_S$ ,  $B$ : realización del sistema en forma material (personal, electrónica, información estática).

Debajo están representadas las relaciones entre S y B:

	S	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	
S	SS	SB <sub>1</sub>	SB <sub>2</sub>	SB <sub>3</sub>	SB <sub>4</sub>	SB <sub>5</sub>	
B <sub>1</sub>	B <sub>1</sub> S	B <sub>1</sub> B <sub>1</sub>	B <sub>1</sub> B <sub>2</sub>	B <sub>1</sub> B <sub>3</sub>	B <sub>1</sub> B <sub>4</sub>	B <sub>1</sub> B <sub>5</sub>	
SB  = B <sub>2</sub>	B <sub>2</sub> S	B <sub>2</sub> B <sub>1</sub>	B <sub>2</sub> B <sub>2</sub>	B <sub>2</sub> B <sub>3</sub>	B <sub>2</sub> B <sub>4</sub>	B <sub>2</sub> B <sub>5</sub>	... (2),
B <sub>3</sub>	B <sub>3</sub> S	B <sub>3</sub> B <sub>1</sub>	B <sub>3</sub> B <sub>2</sub>	B <sub>3</sub> B <sub>3</sub>	B <sub>3</sub> B <sub>4</sub>	B <sub>3</sub> B <sub>5</sub>	
B <sub>4</sub>	B <sub>4</sub> S	B <sub>4</sub> B <sub>1</sub>	B <sub>4</sub> B <sub>2</sub>	B <sub>4</sub> B <sub>3</sub>	B <sub>4</sub> B <sub>4</sub>	B <sub>4</sub> B <sub>5</sub>	
B <sub>5</sub>	B <sub>5</sub> S	B <sub>5</sub> B <sub>1</sub>	B <sub>5</sub> B <sub>2</sub>	B <sub>5</sub> B <sub>3</sub>	B <sub>5</sub> B <sub>4</sub>	B <sub>5</sub> B <sub>5</sub>	

o en forma de una matriz de bloques:

$$|SB| = \begin{vmatrix} SS & SB \\ BS & BB \end{vmatrix} \dots (3),$$

donde:  $B_1S$ : es el funcionamiento intelectual del personal en el sistema;  $SB_1$ : es la distribución de las tareas del sistema en el personal;  $B_2S$ : es el funcionamiento físico del personal (respecto del teclado, la pantalla, etc.);  $SB_2$ : es la distribución de las operaciones del sistema entre el personal (visión, sonido, mantenimiento, etc.);  $B_3S$ : es la transformación de procesos electrónicos en procesos informáticos;  $SB_3$ : es el inverso a  $B_3S$ ;  $B_4S$ : es la participación del sistema operativo en la resolución de las tareas informáticas;  $SB_4$ : es la inicialización del sistema operativo;  $B_5S$ : es la utilización de la información estática (reglas, constantes, bases de datos) en la resolución de las tareas informáticas;  $SB_5$ : es la transformación de la información dinámica a la forma estática o demanda de información; ( $B_1B_1, \dots, B_4B_4$ ): es la interface entre personal y computadora.

Se pueden delimitar las partes  $V_1, V_2, \dots, V_\infty$  respecto del medio exterior de acuerdo a la cantidad e importancia de las mismas. Análogamente, se pueden representar las relaciones entre las partes materiales del sistema ( $B_1, \dots, B_5$ ) y las partes del medio exterior. Si  $V_2$ , son los productores de hardware, entonces  $B_3V_2$  y  $V_2B_3$  es la instalación e eliminación del hardware. Si  $V_3$  son las empresas desarrolladoras de software, entonces  $B_4V_3$  y  $V_3B_4$  es la instalación, modificación y eliminación del software. Si  $V_5$  son las universidades que preparan el personal, entonces  $V_5B_1, V_5B_2$  es el personal contratado, etc. Utilizando vectores y matrices se puede representar los sistemas en forma detallada.

### El análisis de los sistemas de programación [3, 4]

La función (F) del sistema de programación puede ser representada por la ecuación (1), donde:  $I_E$ : son los datos de una tarea a resolver e  $I_S$  es la tarea resuelta. Se puede simplificar el modelo general de la figura 1, representando el sistema de programación y las partes principales de la teoría de sistemas de programación como un conjunto de partes del modelo general:

$$\langle I_S = F(I_E); B_1-B_5; BB; V_1, V_2, \dots, V_\infty; \text{etc} \rangle \dots (2),$$

donde  $I_S \sim SV$ ;  $I_E \sim VS$ ; F: es una función del S; BB: es la correlación entre las propiedades de  $B_1$ ;  $B_5$ ;  $V_1$ : es el subsistema económico  $V_2$ : los gobiernos,  $V_3$ : la ciencia, etc.

La función dinámica (F) en los sistemas discretos se puede reemplazar por una secuencia de subfunciones (3) o bien se la puede representar también en forma de un grafo.

$$F \sim F_1 \rightarrow F_2 \rightarrow F_3 \rightarrow \dots \rightarrow F_n \dots (3),$$

Ahora, se puede resolver el problema de la elaboración de una función F de programación (por ejemplo, un algoritmo) como un problema de estructuras de datos y organización de procedimientos en un lenguaje de programación.

El análisis de los lenguajes concretos tales como: Ada, Natural, C++, LISP, PROLOG, permitió obtener algunos resultados y seleccionar algunas particularidades de esos lenguajes. En Ada, por ejemplo, la particularidad del medio exterior es que la creación de dicho lenguaje ha sido promovida por el Departamento de Defensa de los estados Unidos, o sea un ente estatal; en Natural, las formas flexibles de mantenimiento de grandes volúmenes de datos; en C++ la organización del programa en forma del grafo orientado, utilizando herencia, polimorfismo, sobrecarga, encapsulamiento; en LISP la organización funcional de procedimientos como  $F_n ( \dots ( F_2 ( F_1 ( I_E ) ) \dots )$  y en PROLOG la organización de procedimientos con base en reglas (condición habilitadora  $i \rightarrow$  acción  $i$  ).

### El pronóstico de evolución de los sistemas de computación [5]

En los sistemas de computación (Figura 1) se puede seleccionar una característica (z) del sistema, como relación entre cantidad de operaciones de hardware y todas las operaciones del sistema. Los sistemas de computación tienen algunas regularidades de evolución como se expresa debajo::

$$0 \leq z \leq 1 \dots (4); z = z(t-t_0), t: \text{ tiempo}, t_0: \text{ el momento de comienzo de la evolución} \dots (5);$$

$$z(t-t_0)_{t=t_0} = 0 \dots (6); z(t-t_0)_{t \rightarrow \infty} \rightarrow 1 \dots (7);$$

$$T = (t_0, \infty), T: \text{ intervalo de la evolución} \dots (8);$$

$$z'(t-t_0) > 0, t \in T \dots (9); z'(t-t_0)_{t=t_0} = 0 \dots (10); z''(t-t_0)_{t=t_0} > 0 \dots (11);$$

$$\Delta V_{t \rightarrow \infty} \rightarrow 0 \text{ (nuevos recursos de producción de la base B)} \dots (12);$$

$$\Delta W_{t \rightarrow \infty} \rightarrow 0 \text{ (nuevas potencias de producción de la base B)} \dots (13);$$

$$z'(t-t_0)_{t \rightarrow \infty} \rightarrow 0 \dots (14); z''(t-t_0)_{t \rightarrow \infty} \rightarrow 0 \dots (15).$$

Todos estas regularidades de evolución se pueden representar en forma de una función:

$$z(t-t_0) = [\lambda^2 (t-t_0)^2] / [\lambda^2 (t-t_0)^2 + 1] \dots (16),$$

donde  $\lambda$  (lambda) es el *parámetro de evolución*. Por aplicación de la ecuación (16) se podrá obtener el pronóstico concreto de evolución del sistema.

### Conclusiones

En trabajos anteriores [1] se presentó el análisis de sistemas como una nueva ciencia, que estudiaba las transformaciones de las propiedades de la materia, por lo cual pareció pertinente aplicar esta nueva visión de la teoría general de sistemas.

Ahora, a fin de confirmar los conceptos teóricos en la práctica, es deseable analizar sistemas de tipos diferentes tales como los informáticos, económicos, químicos, físicos, tecnológicos, etc.

En este trabajo se presenta la aplicación de los nuevos conceptos teóricos en sistemas informáticos: de computación (en los estadios estáticos y en la evolución cíclica) y de programación (en los estadios estáticos y dinámicos), que luego se extenderá a otros sistemas.

### Bibliografía

- [1]. Zubenko Y.D., Donnataria C., Lage F.J., Cataldy Z. "El análisis de sistemas como nueva ciencia: Estudiando las transformaciones de las propiedades de la materia". Proceedings del VI Congreso Internacional de Ciencias de Computación, Ushuaia, Argentina, 2000.
- [2]. Zubenko Y.D., Nosach A.K., Sharapov A.D. "Management con el análisis de sistemas como fundamento". Universidad Estatal Técnica de Donetsk, Ucrania, 1998.
- [3]. Ojeda E., Tibaldi E., Zubenko Y.D., Lage F.J., Cataldi Z. "Bases para una teoría de sistemas de programación" (en prensa), 2001.

- [4]. Ojeda E., Tibaldi E., Zubenko Y.D., Lage F.J., Cataldi Z. “*Aplicaciones de la teoría de los sistemas de programación*” (en prensa), 2001.
- [5]. Zubenko Y.D., Iliin A.A. “*Optimización de soluciones de problemas de producción*”. Libro Editado por “Statistika”, Moscú, 1977.