

SISTEMAS EVOLUTIVOS: NUEVO PARADIGMA DE LA INFORMÁTICA

*Fernando Galindo Soria**

Resumen

En este trabajo se presenta una introducción a los Sistemas Evolutivos (sistemas capaces de construir y mantener en tiempo real una imagen del ambiente que los rodea, como un ser que construye su propia imagen de la realidad y la utiliza para interactuar con su entorno), incluyendo su descripción general y presentación de algunos de los métodos y algoritmos más usados en la construcción de este tipo de sistemas principalmente la Arquitectura del Sistema Evolutivo y su estructura Lingüística, se describe el Mecanismo Léxico, el Mecanismo Sintáctico, las principales herramientas de Inferencia Gramatical, el Mecanismo Semántico y las principales herramientas de Diálogo y de asociación de significado.

Palabras Clave: Sistemas Evolutivos, Lingüística Matemática, Reconocimiento de Formas, Sistemas de Información, Base de Datos, Inferencia Gramatical, Herramientas Automatizadas

INTRODUCCIÓN

En éste trabajo se describe en forma general la arquitectura y aplicaciones de los Sistemas Evolutivos, donde un Sistema Evolutivo es una herramienta automatizada capaz de construir una imagen del ambiente que la rodea y a partir de esta imagen resolver problemas, como un niño que construye su propia imagen de la realidad y la utiliza para interactuar con su entorno.

* Fernando Galindo Soria escribió este trabajo en 1990 siendo profesor-investigador en la Licenciatura en Ciencias de la Informática Sección de Graduados e investigación UPIICSA-IPN

Como primer punto se describe en forma general, cuales son los principales problemas de la Informática actual y la arquitectura general de los Sistemas de Información y a partir de este punto se sigue una secuencia en espiral donde en cada vuelta de la secuencia se replantea la arquitectura de los sistemas de información hasta terminar en la conclusión con la arquitectura generalizada de los Sistemas Evolutivos.

I INTRODUCCIÓN A LOS SISTEMAS EVOLUTIVOS

Uno de los problemas más graves de la Informática actual se presenta por su poca capacidad para modelar en tiempo real los fenómenos que ocurren en la realidad, ya que es común que cuando un Sistema de Información: Nómina, Compilador, Reconocedor de Imágenes, Sistema de Inventarios, Sistema Experto de Diagnóstico Médico, etc., se libera ya prácticamente es obsoleto, ya sea porque:

- 1) *El problema modelado se modificó.*
- 2) *Porque el modelo no cubrió los aspectos esenciales.*
- 3) *O simplemente la información y el conocimiento que se tiene sobre el tema ha quedado rebasado por algún nuevo dato o hecho conocido previamente.*

En general se pueden plantear tres grandes problemas que ayudan a la rápida obsolescencia de los sistemas de información y que obligan a realizar en forma continua lo que se conoce como mantenimiento adaptativo (aquel que se realiza para lograr que el sistema mantenga una imagen lo más cercana al ambiente que se requiere modelar) o por el contrario obliga a los usuarios a lidiar con una herramienta cada vez menos poderosa y más alejada de la realidad que aparentemente está modelando.

El primer problema surge cuando se desarrollan sistemas que modelan fenómenos altamente cambiantes, como por ejemplo los sistemas de algo tan cotidiano como la Nómina, donde prácticamente no se ha terminado de desarrollar cuando ya se

tiene que modificar (y no es por mal desarrollo, sino simplemente porque se está tratando de modelar un problema que cambia prácticamente por decreto y en el que es difícil, establecer los patrones de cambiado).

El segundo problema se presenta cuando se desarrollan sistemas que modelan fenómenos para los cuales no existe una regla o patrón ya establecido y bien manejado y del cual se pueda tener un algoritmo en forma relativamente sencilla, tal es el caso de muchos de los problemas actuales de la Inteligencia Artificial. por ejemplo en el reconocimiento de patrones se ha desarrollado gran cantidad de métodos y algoritmos pero sin embargo los problemas no resueltos son cada día mayores, por lo que los Sistemas de Información orientados al reconocimiento de patrones reflejan un modelo de la realidad normalmente restringido.

El tercer problema es un cuello de botella de la Informática que se presenta cuando los sistemas funcionando son incapaces de reflejar la realidad por que los datos y hechos de ésta no son directamente accesibles por el sistema ya sea porque son difíciles de obtener (por ejemplo, topografía detallada del terreno donde se construirá una carretera) o porque cambian tan rápidamente que los métodos tradicionales de captura no permiten mantenerlos actualizados (por ejemplo, los datos clínicos de un paciente de terapia intensiva).

Es por lo anterior que es necesario replantear el enfoque utilizado para resolver problemas en Informática (representado por áreas como el Desarrollo de Sistemas, la Ingeniería de Software y la Ingeniería de Conocimiento) en el cual la tendencia es a la construcción de sistemas estáticos e Incapaces de automantenerse y buscar métodos y herramientas incapaces de recrear en forma continua su imagen de la realidad o del problema a resolver.

Es dentro de este contexto donde surge el concepto de Sistema Evolutivo como un sistema capaz de crear su propia imagen de la realidad y utilizarla para resolver problemas y en su momento mantener actualizada esta imagen mediante un proceso continuo de actualización.

II DEL SISTEMA DE INFORMACIÓN AL SISTEMA EVOLUTIVO

1 RUMBO AL CONSTRUCTOR

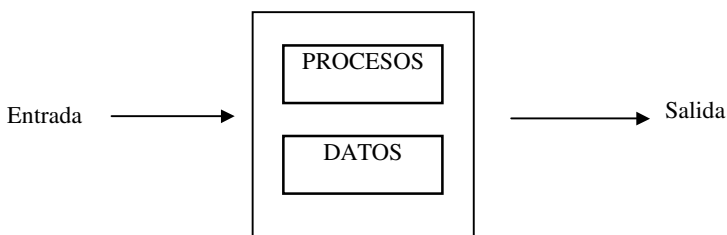
En general se considera que un Sistema de Información o programa de cómputo tiene la arquitectura de la figura 1.



Figura 1. Arquitectura de los sistemas de información en los 50s

Sin embargo este esquema es muy viejo y corresponde a los años 50's y ya para la década de los 60's se manejaban los esquemas más elaborados mostrados en la figura 2.

a. Desarrollo de aplicaciones



b. Compiladores y sistemas operativos

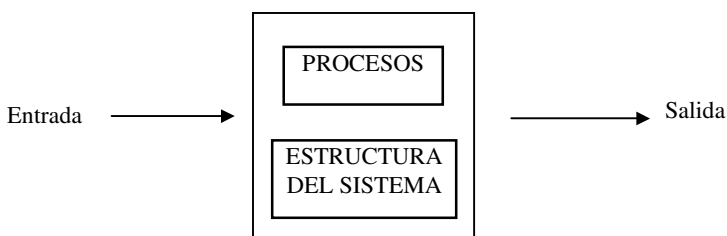


Figura 2. Arquitectura de los sistemas de información en los 60s

El primer esquema se utilizaba principalmente por la gente orientada al desarrollo de aplicaciones y el segundo por las personas dedicadas a desarrollar herramientas de cómputo como Compiladores y Sistemas Operativos, y ya para finales de los 70's se manejaba un modelo generalizado en el cual se considera que cualquier sistema de información tiene la arquitectura de la figura 3.

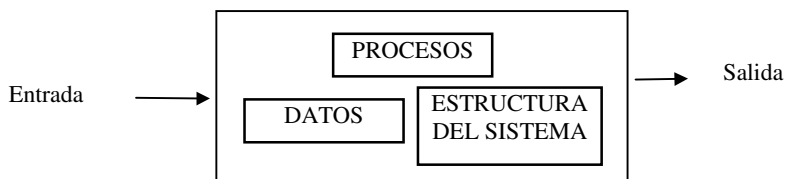


Figura 3. Arquitectura de los sistemas de información en los 70s

Este esquema integra los enfoques anteriores y enfatiza la necesidad de desarrollar los sistemas tomando en cuenta las tres componentes y no solo los procesos, sin embargo, el desconocimiento o el hecho de no tomar en cuenta este esquema cuando se desarrollan sistemas es una de las causas principales por las que los sistemas se vuelven altamente estáticos y difíciles de mantener ya que en los sistemas y programas tradicionales las tres componentes se encuentran revueltas, por lo que, un cambio "pequeño" en los datos o procesos o en el orden de atacar un problema ocasiona que prácticamente se tenga que volver a programar todo; por otro lado si se desarrolla el sistema de tal manera que los datos queden en un lado, los procesos en otro y finalmente la estructura del sistema en otro, el proceso de actualización puede ser relativamente fácil (un caso particular de este enfoque es el del desarrollo de Bases de Datos).

Por lo que se considera que una característica fundamental que se debe buscar cuando se desarrolla un sistema de información es la de que exista una Independencia Relativa o sea que los datos, procesos y estructura del sistema queden separados y únicamente exista la relación mínima necesaria entre las tres componentes.

La Independencia Relativa es una de las características distintiva del método de desarrollo basado en Base de Datos y en este documento simplemente se extiende al esquema general de Sistemas de Información.

Otra característica presente en el método de base de datos que se puede extender al desarrollo de sistemas es el hecho de que en el enfoque de base de datos se distinguen explícitamente tres niveles, en el primero se encuentran los datos en sí (nombres propios, edades concretas, direcciones específicas, etc.); en el segundo se tiene la estructura de datos específica donde se almacenan los datos (archivos, registros y campos de la base de datos que se está manejando) y en el tercer nivel se tiene un programa que se encarga de construir una estructura de datos particular, a partir de una descripción general de la base de datos, este programa es el constructor de la base de datos.

En realidad una base de datos no tiene al principio datos o estructuras de datos particulares, sino que solo cuenta con el constructor que es capaz de generar múltiples casos particulares.

En el caso de sistemas de información esta idea se puede generalizar, buscando que, más que tener datos, procesos o estructuras particulares de un sistema, se cuente con mecanismos que nos permitan construir los datos, procesos y estructuras a partir de una descripción general del sistema. como se muestra en el esquema de la figura 4.

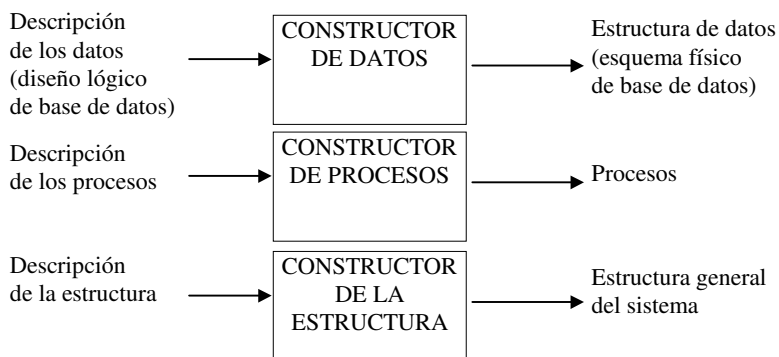


Figura 4. Constructores de datos, procesos y estructura

Estos tres módulos se pueden integrar en el esquema mostrado en la figura 5.

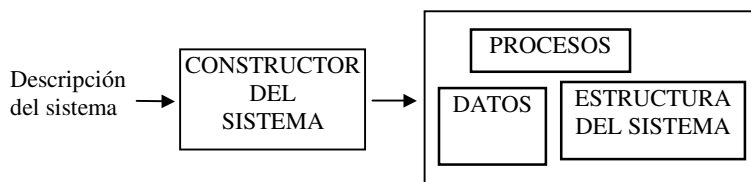


Figura 5. Integración de los constructores en un constructor del sistema

Donde el constructor del sistema es un programa capaz de construir la estructura general de un Sistema de Información a partir de la descripción del sistema, de la misma forma que un constructor de Base de Datos es capaz de construir el esquema físico de una Base de Datos a partir de un Esquema Lógico de Base de Datos.

2 RUMBO A LA ARQUITECTURA DE UN SISTEMA EVOLUTIVO

Si se construye el sistema de información de tal forma que los componentes sean independientes en forma relativa entre sí y se da una interrelación entre el constructor y el sistema de tal forma que cualquier cambio en la descripción del sistema se refleje en tiempo real en el sistema de información, entonces se puede considerar que la imagen de la realidad que tiene el sistema de información es bastante cercana a la que se quiere reflejar. Ahora bien cuando un programa sigue un conjunto de reglas o instrucciones para resolver un problema se dice que es un Programa Deductivo, y a los métodos que construyen programas deductivos se les conoce como métodos de programación deductiva.

El problema de la programación deductiva es que se orienta a desarrollar sistemas fijos difíciles de modificar en tiempo real (ya que las modificaciones involucran reprogramar el sistema), por lo que desde hace tiempo en el área de Sistemas Evolutivos se ha

buscado desarrollar herramientas automatizadas capaces de obtener en forma automática el conjunto de reglas del sistema a partir de ejemplos y descripciones generales de un programa.

A una herramienta o programa capaz de encontrar un conjunto de reglas a partir de ejemplos la conocemos como Mecanismo o Herramienta Inductiva; los mecanismos inductivos fueron de las primeras herramientas utilizadas en el desarrollo de constructores y son de las más generales, el esquema de un constructor basado en una herramienta inductiva es el mostrado en la figura 6.

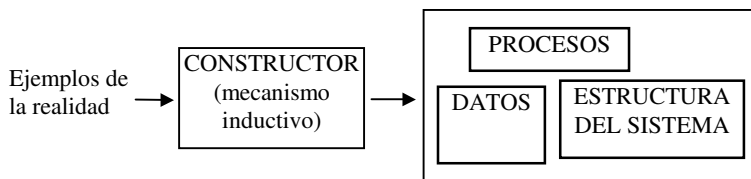


Figura 6. Herramienta inductiva

En el esquema anterior se maneja un mecanismo inductivo para construir una Imagen de la Realidad, sin embargo en esta imagen solo se tienen un conjunto de reglas en términos de procesos, estructura y datos del sistema por lo que es necesario integrar una herramienta capaz de resolver problemas siguiendo esas reglas, o sea un mecanismo de tipo deductivo como se ve en el diagrama de la figura 7.

Para contar con un Sistema Evolutivo lo único que hace falta es dotar a esta herramienta con la capacidad de actualizar su imagen de la realidad en tiempo real con lo que se tiene la arquitectura de la figura 8.

De donde la Arquitectura General de un Sistema Evolutivo es la mostrada en la figura 9.

III ARQUITECTURA

En el punto anterior se vio que con el fin de resolver problemas el sistema evolutivo construye una imagen de la realidad, pero no se indicó como se lleva a cabo este proceso. por lo que en este punto se describirá en forma general los principales métodos y algoritmos utilizados por los Sistemas Evolutivos; para lo cual se partirá de que el lenguaje de comunicación con el exterior (o sea el lenguaje con el que se plantean los problemas al Sistema Evolutivo y con el cual éste responde a los requerimientos) y el lenguaje con el que el Sistema Evolutivo construye su imagen de la realidad son el mismo.

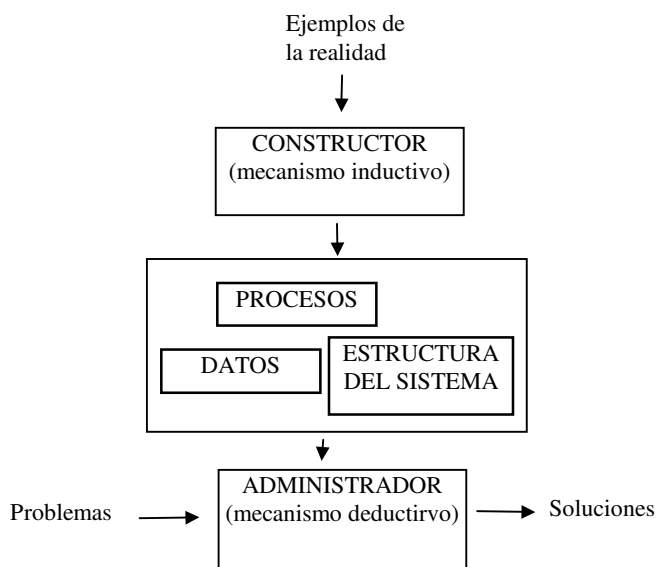
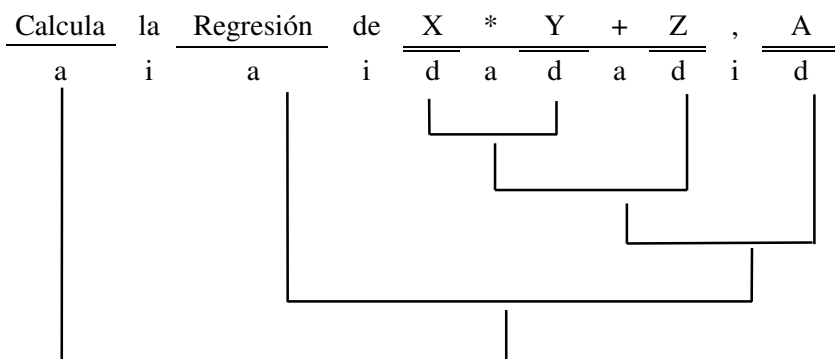


Figura 7. Integración de la componente deductiva

En la actualidad ya existen una gran cantidad de herramientas orientadas al manejo lingüístico de los Sistemas Evolutivos y se basan en que es relativamente fácil encontrar los elementos de un Sistema dentro de las oraciones, por ejemplo en la oración:

Calcula la Regresión de $X * Y + Z$, A

Es relativamente fácil detectar los datos, acciones y estructura:



la estructura se indica con las líneas y representa el orden en que se ejecutan las acciones sobre los datos, los datos tienen el tipo **d**, las acciones el tipo **a** y las palabras no relevantes llevan **i** (ignora).

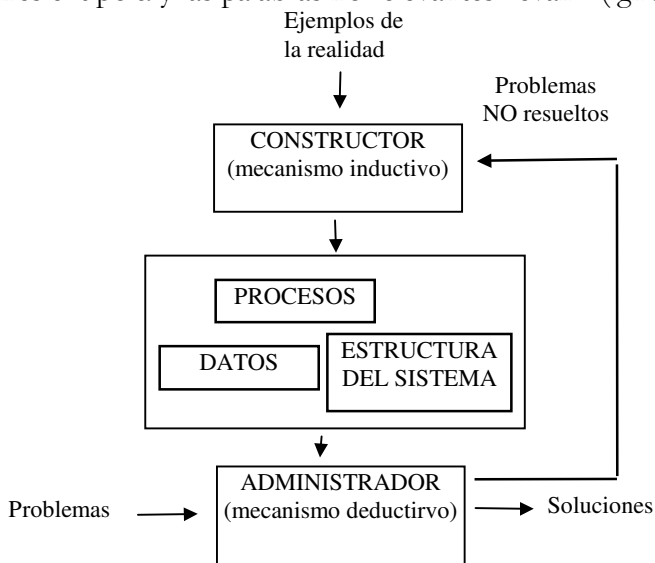


Figura 8. Retroalimentación para actualizar la imagen de la realidad

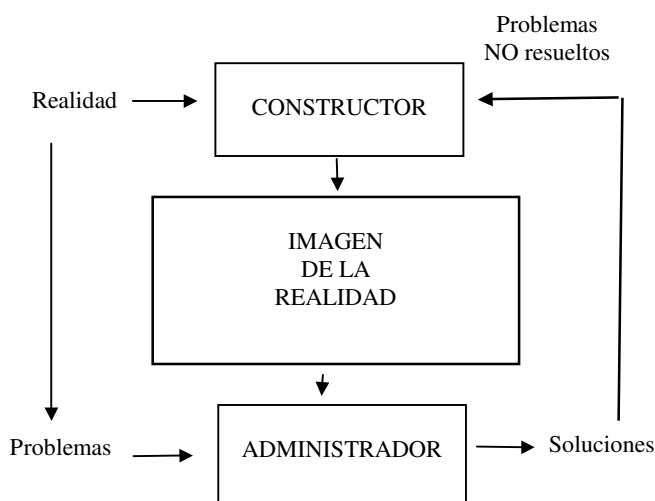


Figura 9. Arquitectura General de un Sistema Evolutivo

Entonces un Sistema Evolutivo debe tener la capacidad de encontrar los componentes del sistema de información a partir del lenguaje utilizado en el área problema y para lograrlo muchos sistemas evolutivos tienen la arquitectura de la figura 10.

Donde el Constructor/Analizador Léxico es el encargado de encontrar cada una de las unidades léxicas que componen una oración, de que tipo es cada unidad (por ejemplo dato u oración) y la estructura general de la oración conocida como oración canónica (formada por la concatenación de los tipos de las unidades léxicas).

A partir de ahí el Constructor/Analizador Sintáctico aplicando métodos de inferencia gramatical (métodos orientados a encontrar la gramática de un lenguaje a partir de ejemplos de oraciones del lenguaje) encuentra la estructura del sistema y finalmente el Constructor/Analizador Semántico, encuentra significado de la oración utilizando mecanismos de dialogo para encontrar el significado de los elementos desconocidos.

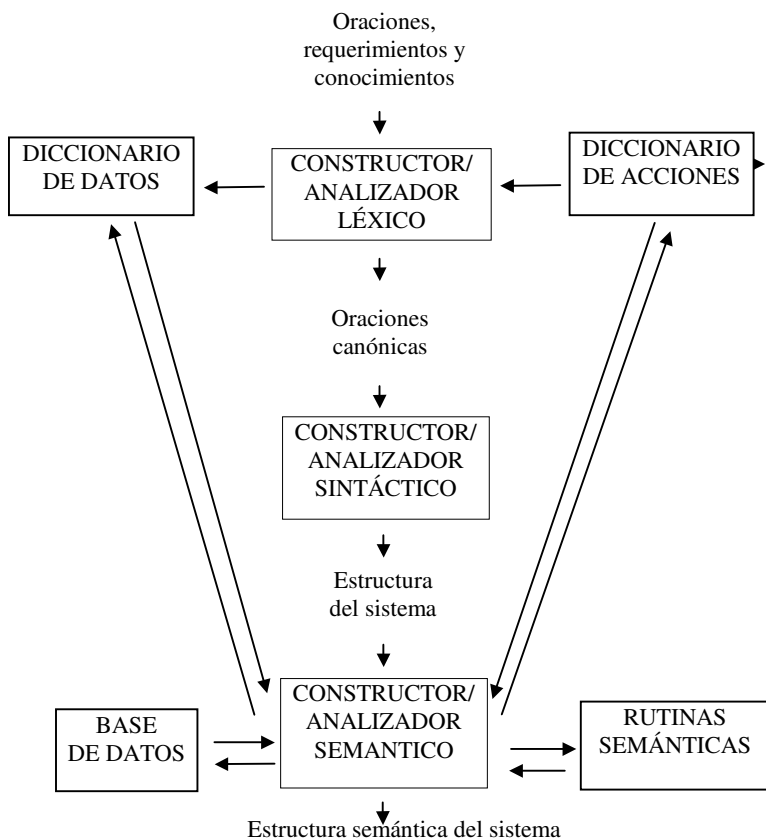


Figura 10. Arquitectura de un sistema evolutivo

A continuación se describen algunos de los principales métodos utilizados por cada uno de los constructores/analizadores.

1 CONSTRUCTOR/ANALIZADOR LÉXICO

El Constructor/Analizador Léxico es un programa que recibe como entrada una oración en algún lenguaje y obtiene las unidades léxicas presentes en la oración, el tipo de cada unidad léxica y la estructura general u oración canónica.

Por ejemplo, dada la oración

Calcula la Regresión de $X * Y$, Z

se encuentran las Unidades Léxicas mostradas en la tabla 1.

Calcula	a, acción
la	i, ignora
Regresión	a
de	i
X	d, dato
*	a
Y	d
,	i
Z	d

Tabla 1. Unidades Léxicas

Originalmente el Constructor/Analizador Léxico no tiene ningún conocimiento acerca del lenguaje a utilizar y solamente cuenta con la capacidad para encontrar los diferentes tipos de unidades léxicas de un lenguaje para lo cual utiliza entre otros los siguientes métodos "triviales":

- 1) *Unidades predefinidas*: Es el método más usado y el más antiguo, consiste básicamente en almacenar previamente la tabla de unidades léxicas con las palabras que se utilizan en el sistema indicando de que tipo son. Con la característica de que en un sistema evolutivo esta tabla es creciente (o sea que originalmente la tabla tiene las unidades léxicas mas usadas en el sistema y se cuenta con la capacidad de almacenar nuevas unidades conforme se van detectando).
- 2) *Caracteres especiales*: El mecanismo anterior tiene la limitante de que si se requiere utilizar una palabra no catalogada el sistema tiene que preguntar y esto al principio puede ser tedioso, por lo que, cuando se construyeron los primeros ensambladores y compiladores se utilizó el truco de manejar ciertos caracteres o posiciones específicas dentro del programa para que el sistema asignara las unidades léxicas. Por ejemplo en un ensamblador dado, la instrucción:

:Etiqueta .Mem #3 ;Coloca 3 en la etiqueta

es fácilmente analizable si se tiene la convención de que:

: Indica Tipo de Etiqueta

. Indica Tipo de Instrucción

Indica tipo de Dato

; Indica Tipo de Comentario

En general esta idea se puede utilizar para catalogar nuevas unidades Léxicas.

3) *Mecanismo de Dialogo*: En un Sistema Evolutivo muchas veces se desconoce hasta el tipo de unidades léxicas que conforman el lenguaje, por lo que, se ha visto que un mecanismo muy general es aquel que permite catalogar en tiempo real las nuevas unidades Léxicas. para lo cual se maneja el esquema de la figura 11.

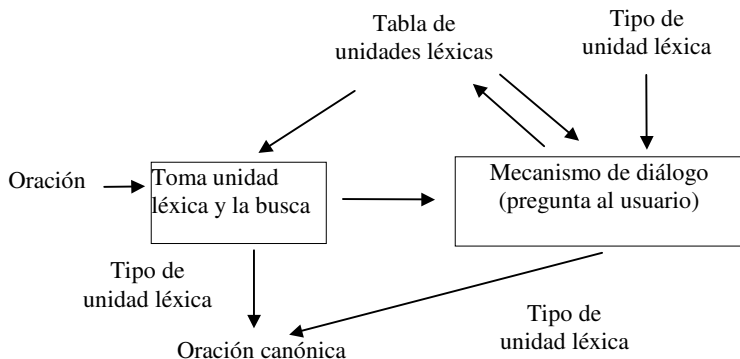


Figura 11. Mecanismo de diálogo

Este mecanismo "trivial" es extremadamente fuerte y refleja el proceso que sigue un niño cuando desconoce una palabra.

4) *Manejo de Contexto*: Los anteriores métodos tienen el inconveniente de que todo se le tiene que dar al sistema y no

utilizan mucha de la información ya obtenida. En el momento que ya se han manejado varias oraciones se pueden empezar a detectar algunos patrones que se pueden utilizar para encontrar el tipo de nuevas unidades léxicas. Por ejemplo, si después de catalogar 10 oraciones se encontró que la estructura:

a i a i d

se repite 4 veces y el sistema recibe la oración

Calcula la media de A

donde ya conoce el tipo de las unidades

calcula (a)

la (i)

de (i)

A (d)

donde a=acción, d=dato, i=ignora; entonces la oración canónica tiene la forma:

a i ? i d

Por lo que el sistema propone que el elemento desconocido

Media es de tipo a.

Los anteriores métodos "triviales" al combinarse son extremadamente poderosos y en su momento con una cantidad relativamente pequeña de ejemplos son capaces de encontrar el conjunto base de palabras manejadas por un área problema.

Existen otros métodos más poderosos pero normalmente requieren del apoyo de los mecanismos Sintáctico y Semántico, estos últimos normalmente no se utilizan en los prototipos de Sistemas Evolutivos.

2 CONSTRUCTOR/ANALIZADOR SINTÁCTICO

A partir de los resultados del Constructor/Analizador Léxico el Constructor/Analizador Sintáctico encuentra la estructura del sistema para lo cual, toma como entrada el conjunto de oraciones canónicas y genera la Estructura.

Este mecanismo es tal vez el componente mas importante de un Sistema Evolutivo ya que es el responsable de encontrar las reglas generales o patrones de estructura del sistema y para lograrlo utiliza normalmente métodos de la inferencia gramatical.

La Inferencia Gramatical es una herramienta de la lingüística Matemática utilizada originalmente en el área de reconocimiento de Patrones y que posteriormente se ha extendido y usado masivamente en la construcción de Sistemas Evolutivos.

El problema que ataca la Inferencia Gramatical consistir básicamente en encontrar la Gramática (Estructura) que describe a un lenguaje dado a partir de ejemplos de oraciones del Lenguaje, figura 12.



Figura 12. Inferencia gramatical

En sus inicios se desarrollaron un conjunto de algoritmos orientados a resolver problemas específicos y en la mayoría de los casos eran métodos difíciles de entender y mas difíciles de programar, pero conforme se empezó a atacar el problema para construir Sistemas Evolutivos se fueron encontrando nuevos métodos y generalizando el problema, por lo que, en la actualidad se ha encontrado que prácticamente los métodos de inferencia gramatical se basan en las operaciones básicas de:

Factorización, Distribución y Recursividad por lo que a continuación se explicaran estas operaciones.

3 FACTORIZACIÓN LINGÜÍSTICA

En su forma mas sencilla la factorización lingüística lo que pretende precisamente es encontrar los factores comunes en diferentes oraciones.

Por ejemplo si se tienen las siguientes oraciones canónicas (S indica el primer nivel de factorización).

$$\begin{array}{l} S \rightarrow a i b i a d o f l \\ \quad a i b i d o f l \\ \quad \quad a i d a d f \end{array}$$

factorizando la cadena **a i** que es común a las tres oraciones de S

$$\begin{array}{l} S \rightarrow a i X \\ X \rightarrow b i a d o f l \\ \quad b i d o f l \\ \quad \quad d a d f \end{array}$$

Factorizando **b i** en las dos primeras oraciones de X queda

$$\begin{array}{l} S \rightarrow a i X \\ X \rightarrow b i Y l \\ \quad a d o f \\ Y \rightarrow a d o f l \\ \quad d o f \end{array}$$

Si se observa, lo único que hace el proceso de factorización es sacar los factores comunes (que se repiten) de las diferentes oraciones.

4 RECURSIVIDAD LINGÜÍSTICA

El proceso de recursividad lingüística también busca un conjunto de elementos comunes dentro de las oraciones pero con la diferencia de que se buscan cadenas de elementos que se repiten periódicamente y en forma consecutiva mas de cierto número mínimo de veces (normalmente tres o más veces) y se asume que esa cadena se puede repetir tantas veces como se quiera.

Por ejemplo en la cadena

$$S \rightarrow a \underline{b\ c} \underline{b\ c} \underline{b\ c} \underline{b\ c} f$$

Los elementos b c se repiten consecutivamente 4 veces por lo que se asume que se pueden repetir tantas veces como se desee y esto se expresa introduciendo un elemento auxiliar y haciéndolo recursivo.

$$\begin{aligned} S &\rightarrow a X f \\ X &\rightarrow \underline{b\ c} \underline{b\ c} \underline{b\ c} \underline{b\ c} \\ &\quad \Downarrow \\ S &\rightarrow a X f \\ X &\rightarrow b\ c\ X \end{aligned}$$

Por ejemplo si se tiene

$$S \rightarrow a i a d a d a d a d f$$

Introduciendo recursividad sobre a d queda:

$$\begin{aligned} S &\rightarrow a i X f \\ X &\rightarrow a d a d a d a d \\ &\quad \Downarrow \\ S &\rightarrow a i X f \\ X &\rightarrow a d X \end{aligned}$$

El proceso de introducir recursividad es un mecanismo extremadamente poderoso ya que permite generalizar una secuencia de repetición y por el otro lado es un mecanismo peligroso ya que se puede generalizar mas de lo debido.

En general, el Constructor/Analizador obtiene la estructura de un sistema a partir de las oraciones del lenguaje y utilizando en forma combinada la recursividad, distribución y factorización.

5 CONSTRUCTOR/ANALIZADOR SEMÁNTICO

Esta componente del Sistema Evolutivo es la encargada de asociar el significado a los elementos y estructura del sistema y se basa principalmente en un mecanismo de dialogo y en la construcción

de relaciones entre los elementos utilizando diferentes mecanismos de captación de la realidad (en la actualidad utiliza principalmente imágenes y lenguaje escrito).

El mecanismo de Diálogo fue originalmente propuesto por José Luis Díaz Salas en 1987 y posteriormente ha sido complementado por otros investigadores y consiste básicamente en un proceso recursivo en el cual cada que el sistema Evolutivo detecta una palabra u oración para la que no encuentra significado simplemente pregunta y espera una respuesta en el mismo lenguaje.

Por ejemplo si se tiene la oración:

calcula la Media de A

y la computadora no sabe lo que significa la palabra Media, pregunta, por lo que se puede establecer un dialogo del siguiente tipo:

Usuario: *Calcula la media de A*

Computadora: *¿Cómo se obtiene la Media?*

Usuario: *Suma los elementos de A y divídelos entre el número de elementos*

Este mecanismo ha sido modificado con el fin de permitir que se definan Sinónimos o unos elementos en término de otros.

CONCLUSIÓN

El área de los Sistemas Evolutivos es muy extensa y en este documento sólo se presentó una introducción al tema ya que desde 1985 se han desarrollado gran cantidad de sistemas de este tipo principalmente a nivel de prototipos, pero ya en la actualidad varias empresas y entidades públicas están construyendo sus propios Sistemas Evolutivos y al menos en un caso se está trabajando en una herramienta de mercado, por lo que ésta es una invitación a que más investigadores e instituciones se sumen a este esfuerzo.

ALGUNAS PUBLICACIONES RELACIONADAS CON SISTEMAS EVOLUTIVOS

Construcción de Sistemas Evolutivos

De la Biblioteca a la Memoria Automatizada

Representación de Conocimiento

Programación Dirigida por Sintaxis

Inferencia Gramatical

Introducción a la Lingüística Matemática

Sistema Evolutivo Constructor de Sistemas Expertos

Sistemas Evolutivos de Lenguaje de Trayectorias

Herramientas Generales Para el desarrollo de juegos y Software Educativo

Herramientas Generales para el Desarrollo de Sistemas (Núcleo de un Sistema Evolutivo)