

UN DICCIONARIO DE DATOS PARA UN SISTEMA DE APOYO A LA TOMA DE DECISIONES.

S. ABAD*

RESUMEN

El objetivo de este trabajo es el diseño de un Diccionario de Datos "no estructurado" que funcione como intermediario entre el diseñador de un Sistema de apoyo a la toma de decisiones (DSS: Decision Support System) y su base de datos. El diccionario brinda apoyo en todas las fases del ciclo de vida de la base de datos (análisis, diseño y mantenimiento). La característica de no estructurado lo hace independiente de los modelos conocidos: relacional, jerárquico y de redes.

ABSTRACT

The purpose of this article is to design a "loosely structured" data dictionary to operate as an intermediary between a Decision Support System (DSS) and the DSS data base. This dictionary is a supporting tool throughout all the phases in the database life cycle. The characteristic of being loosely structured makes the dictionary independent from the other known models: relational, hierarchical and network.

* Ingeniero en Computación (USB, 1980), Magister en Ciencias de la Computación (USB, 1983); base de datos, ingeniería de software, sistemas de información, sistemas de operación; Universidad Simón Bolívar - Departamento de Matemáticas y Ciencias de la Computación. Apartado Postal 80659, CARACAS - VENEZUELA. Telfns: 962-1101 al 09 ext. 7191 y 986-4726.

1. INTRODUCCION.

La utilización de los computadores en el desarrollo de aplicaciones ha atravesado varias etapas. En la década de los 50, los computadores se dedicaban al desarrollo de Sistemas EDP (Electronic Data Processing); en los años 60 y 70, el énfasis se hizo en los Sistemas de Información Gerenciales (Management Information Systems). En los años 80 se habla de los Sistemas de Apoyo a la Toma de Decisiones (DSS: Decision Support Systems). Estos últimos surgieron de la combinación de las disciplinas de Investigación de Operaciones y Sistemas de Información, y hoy en día convergen en los DSS, además de las dos áreas mencionadas, la de la Inteligencia Artificial, la de Lenguajes y la de Base de Datos. Con los DSS se trata de proveer información a los niveles gerenciales de una empresa, para ayudarlos en su proceso de toma de decisiones, abriéndoles la posibilidad de modelar ellos mismos sus problemas, a través de lenguajes de alto nivel. Una metodología que ha dado buenos resultados en el desarrollo de los DSS, es la del Diseño Incremental y evolutivo. De hecho, hoy en día, tal metodología caracteriza la esencia de los DSS. Se pretende con esta metodología, llevar a cabo todas las fases del desarrollo del sistema (análisis, diseño e implantación) en paralelo. De esta manera, el uso del sistema ayuda a su desarrollo, con lo cual un DSS nunca se culmina, por el contrario, constantemente se le hacen modificaciones para mejorarlo.

Los usuarios finales de los DSS son los gerentes, para que ellos puedan obtener mayor provecho de la información para las decisiones, es vital proveerles una interacción "amigable" y directa con el DSS. Esto se puede lograr diseñando lenguajes de alto nivel, los cuales les permitan expresar clara y precisamente, sus necesidades de información. En este aspecto intervienen las áreas de inteligencia artificial y lenguajes de programación.

Por otra parte, los datos que apoyan las decisiones en un DSS pueden ser muy diversos y se requiere de facilidades de recuperación cruzada de información de varias fuentes de datos. En este aspecto son de gran utilidad los Sistemas de Base de Datos. Sin embargo, con el enfoque del diseño evolutivo y el grado de incertidumbre presente en la etapa de análisis, la base de datos de apoyo al DSS, no tiene estructura, es un ente difuso y en crecimiento. En consecuencia, es necesario proveer mecanismos de manejo de esa base de datos "no estructurada". Como complemento de la base de datos, una herramienta muy útil y poderosa para el control y la administración de los datos, es un Diccionario de Datos (DD) (referencias 1, 3 y 8). En el DD de un DSS se debe considerar la no estructuración de los datos que se describen.

La base de datos para un sistema de apoyo a la toma de decisiones, y el diccionario de datos correspondiente, evolucionan a medida que se desarrolla el sistema. En tal evolución se pueden agregar o eliminar elementos del DD y se pueden modificar las interrelaciones entre los mismos. Bajo estas condiciones, el DD ofrece mas facilidades en la medida que dispone de una estructura flexible para adaptarse a los cambios. Tal

flexibilidad requiere de un mecanismo apropiado para recuperar y actualizar información. Motro en [4] propone un método para recuperar información en bases de datos "no estructuradas", llamado "browsing" (hojeada). De acuerdo con este método, se realiza una búsqueda exploratoria sobre la base de datos, sin suponer conocida su organización, mas aún, sin que tenga que existir alguna. Parte de las ideas expresadas por Motro en el artículo mencionado, se utilizan como base para construir el DD propuesto.

En el presente trabajo se describe el diseño de un Diccionario de Datos para un Sistema de apoyo a la toma de decisiones, tomando en cuenta las características de no estructuración de la base de datos de este último. En la sección 2 se introducen los conceptos en los cuales se apoyan el diseño y la construcción del DD; en la sección 3 se define el diccionario propuesto y en la sección 4 se presentan las conclusiones del trabajo y se plantean algunos problemas que requieren investigación.

2. CONCEPTOS BASICOS.

El objetivo de esta sección es plantear los conceptos utilizados en el diseño del diccionario de datos para un DSS.

El punto de partida es que un diccionario de datos es una base de datos con información acerca de los datos de una base de datos. Para diseñar una base de datos se comienza por establecer un modelo conceptual. Existen varias alternativas para modelar conceptualmente los datos ([2], [5] y [6]), entre ellas, una de las mas conocidas y ampliamente utilizada es la del Modelo E-R (Entity-Relationship) ([2], [7]). De acuerdo con este modelo, los datos se representan con entidades (entes distinguibles) y con interrelaciones entre esas entidades. Los conceptos asociados a este modelo son los siguientes:

ENTIDAD	ente distinguible del cual pueden registrarse datos. Puede ser un evento, una persona, un concepto. Contraejemplo: fósforos.
CONJUNTO ENTIDAD	grupo de entidades similares
ATRIBUTO	aquellas propiedades que caracterizan una entidad. También se le llama elemento de dato.
VALOR DE UN ATRIBUTO	dato contenido en un atributo. Puede ser un valor cuantitativo, cualitativo o descriptivo.
DOMINIO	conjunto de valores. Una entidad particular de un conjunto entidad, tiene asociado a cada atributo, un valor tomado de algun dominio.
CLAVE	Atributo o grupo de atributos cuyos valores identifican sin ambigüedad, cada entidad dentro de un conjunto entidad. Como toda entidad es distinguible, cada conjunto entidad tiene por lo menos una clave.
INTERRELACION	Se define una interrelación entre conjuntos entidad como un conjunto

de listas de elementos de los conjuntos entidad, ordenadas. Sean $E_1, E_2, \dots, E_n$ los conjuntos entidad bajo la interrelación, los elementos de esta última tendrán la forma general: $(e_1, e_2, \dots, e_n)$ donde $e_i \in E_i$ para todo i . Lo más común es que la lista tenga solo dos elementos, pero en general puede tener más de dos.

En el diseño que se propone en este trabajo, se utilizan las entidades para representar las unidades básicas de datos en el diccionario de datos, y se utiliza la noción de hecho ("fact") (definida por Motro en [4]) para representar las interrelaciones entre tales entidades. La definición de hecho es la siguiente:

HECHO es una terna (e_1, r, e_2) donde e_1 y e_2 son entidades y r es el nombre de la interrelación que las asocia.

Si se denomina E al conjunto de todas las entidades e I al conjunto de todos los nombres de interrelación, entonces los hechos son elementos del conjunto $(E \times I \times E)$. Si en la definición de hecho, e_1 y e_2 son conjuntos entidad, al hecho expresado se le llama CONJUNTO HECHO. Cuando se elige un elemento particular de cada uno de los conjuntos entidad que intervienen en el hecho, se obtiene una instancia particular del conjunto hecho.

El conjunto de hechos que expresan interrelaciones entre entidades (podemos considerar a las interrelaciones como entidades, [2]), se puede particionar en dos conjuntos:

Interrelaciones individuales (H_i): contiene las interrelaciones que caracterizan a un conjunto entidad y que son aplicables a todo elemento de él. Por ejemplo, suponga el hecho: (elem-dato, OCUPA, longitud) donde elem-dato y longitud son conjuntos entidad. Una instancia de este hecho es (cédula, OCUPA, 9) donde cédula pertenece al conjunto elem-dato y 9 pertenece al conjunto longitud. Se dice que la interrelación OCUPA es una interrelación individual porque cada elemento de elem-dato, está asociado con un elemento de longitud. En este conjunto de interrelaciones individuales, existen ciertos hechos muy particulares, llamados "hechos totales" los cuales se utilizan para expresar alguna propiedad de los elementos que forman el conjunto entidad. Estos hechos se denotan por (E_i, R, e_m) donde E_i es un conjunto entidad y e_m es una entidad particular. Este hecho se interpreta: todos los elementos de E_i están interrelacionados de acuerdo a R , con el elemento e_m . Por ejemplo, si LONG-9 es un conjunto de todos los elementos de 9 posiciones, el hecho (LONG-9, OCUPA, 9) dice que todos los elementos de LONG-9 ocupan 9 posiciones.

Interrelaciones de clase (H_c): contiene las interrelaciones que caracterizan a un conjunto entidad pero que no son aplicables a cada elemento del conjunto sino al todo. Por ejemplo, (elem-dato, N°-TOTAL, 500) es un hecho que se interpreta: el número total de elementos de datos incluidos en el DD es 500, entonces se dice que la

Existen ciertas interrelaciones que expresan condiciones especiales sobre las entidades que asocian. Entre ellas, dos de las más utilizadas son las de pertenencia y las de generalización.

La interrelación de pertenencia expresa que una de las entidades es una instancia de la otra. A los hechos donde ellas participan se les llama "hechos de pertenencia", y se denotan así: $(e_1, \in, e_2)$ donde e_1 es una instancia de e_2 . Un ejemplo de este tipo de interrelación se encuentra en el hecho siguiente (M.Contreras, $\in$, estudiante) el cual nos dice que la entidad M.Contreras pertenece al conjunto entidad estudiante.

En el tipo de interrelación de generalización, el sentido expresado por una de las entidades es más general que el expresado por la otra entidad en el hecho. Los hechos donde estas interrelaciones participan se llaman "hechos de generalización" y se denotan así: $(e_1, \subset, e_2)$. Por ejemplo, el hecho (renault, $\subset$, carro) es de generalización y significa que todo elemento del conjunto entidad renault, también es un elemento del conjunto entidad carro, pero lo inverso no es cierto.

A partir de los hechos, cualquiera que sea su tipo, se puede deducir cierta información que no está explícita en los mismos. Por ejemplo, si cédula es un elemento del conjunto LONG-9 y todos los elementos de LONG-9 ocupan 9 posiciones, se puede deducir que cédula ocupa 9 posiciones. Llevando estas afirmaciones a hechos, tendríamos:

$(\text{cédula}, \in, \text{LONG-9})$ y $(\text{LONG-9}, \text{OCUPA}, 9)$

de lo cual deberíamos poder concluir el hecho: $(\text{cédula}, \text{OCUPA}, 9)$. Este último hecho no tiene por qué existir explícitamente, pues se puede inferir de los dos primeros. Mas aún, la conclusión obtenida se puede generalizar como:

Si $(X, \in, \text{LONG-9})$ y $(\text{LONG-9}, \text{OCUPA}, 9) \Rightarrow (X, \text{OCUPA}, 9) \quad (1)$

donde X es una variable que puede tomar como valor cualquier elemento del conjunto LONG-9. Los hechos donde intervienen variables se les llama "templates" (patrones).

A una expresión del tipo mostrado en (1) se le llama "regla de inferencia". El tipo de deducción que se utilizó en el ejemplo se puede generalizar aún mas con la siguiente regla:

$\forall r \in I: (s, \in, t) \text{ y } (t, r, u) \Rightarrow (s, r, u)$

Existe un conjunto de reglas de inferencia universales ([4]) que se pueden aplicar al ambiente de base de datos. Algunas de estas reglas surgen de hechos de generalización o de pertenencia y son muy útiles, las que se muestran a continuación son algunas de ellas:

$\forall r \in I: (s, r, t) \text{ y } (s', \subset, s) \Rightarrow (s', r, t)$

$\forall r \in I: (s, r, t) \text{ y } (r, \subset, r') \Rightarrow (s, r', t)$

$\forall r \in I: (s, r, t) \text{ y } (t, \subset, t') \Rightarrow (s, r, t')$

$\forall r \in I: (s, r, t) \text{ y } (s', \in, s) \Rightarrow (s', r, t)$

Un aspecto vital en la vida de una base de datos, es la integridad de los datos. A menudo la porción más engorrosa del desarrollo de programas de aplicación y de las operaciones de mantenimiento, es la del control de la integridad de los datos. Es deseable entonces, disponer de un mecanismo automatizado para verificar tal integridad al efectuar operaciones de actualización de la base de datos. Una vía para implantar tal mecanismo es agregar al DD la información concerniente a las restricciones de integridad que es necesario imponer sobre los datos, y que antes de realizar cualquier actualización, se verifique automáticamente con el DD, que tal actualización no viole la integridad de los datos.

Las restricciones de integridad se pueden expresar con reglas como la siguiente: $(x, \in, \text{cedula}) \Rightarrow (x, \text{MAYOR-QUE}, 0)$. Esta regla se interpreta como que todo elemento del conjunto de valores válidos para la cédula de identidad, debe ser positivo. Estas restricciones, en lo que respecta a forma, son como las reglas de inferencia.

Es oportuno aclarar que con estas construcciones se pueden expresar hechos contradictorios, por ejemplo, si además de la restricción expresada anteriormente, se incluyera la siguiente regla: $(x, \in, \text{cédula}) \Rightarrow (x, \text{MENOR-QUE}, 0)$, se crearía una contradicción puesto que todo elemento del conjunto cédula no puede ser mayor y menor que cero simultáneamente. La contradicción provino de que las interrelaciones MAYOR-QUE y MENOR-QUE no pueden establecerse simultáneamente sobre los mismos elementos, porque son contradictorias. En consecuencia, es útil conocer cuales interrelaciones son contradictorias entre sí, para evitar la existencia de hechos contradictorios en la base de datos y en el diccionario de datos.

Durante el diseño de una base de datos, es de suma relevancia conocer la semántica de los datos, para poder construir una base de datos fiel a la realidad. Por ejemplo se tienen los conjuntos entidad tutor y estudiante; se admite que cada tutor pueda tener varios estudiantes, pero cada estudiante tiene a lo sumo un tutor, ello establece una interrelación 1:n de tutor a estudiante, la cual se podría expresar con la siguiente regla:

$$\begin{aligned} &(x, \in, \text{tutor}) \text{ y } (e, \in, \text{estudiante}) \text{ y} \\ &(x, \text{ES-TUTOR}, e) \text{ y } (z, \in, \text{tutor}) \quad \Rightarrow \quad (x, =, z) \\ &(z, \text{ES-TUTOR}, e) \end{aligned}$$

Sin embargo, como las interrelaciones 1:n, junto con las 1:1 y las n:m son muy comunes en el ambiente de base de datos, es muy útil disponer de una expresión abreviada para representarlas. Una manera de hacerlo, es utilizar un operador llamado TIPO, el cual se aplica a las interrelaciones y les asigna su tipo. La forma general de este operador será: $\text{TIPO}(\text{interrelación}) = \text{valor}$, los valores posibles son: 1:1, 1:n, n:1 y n:m. En nuestro ejemplo, se definiría ES-TUTOR como una interrelación 1:n, y se escribiría como: $\text{TIPO}(\text{ES-TUTOR}) = 1:n$.

Al utilizar este operador se expresa una restricción de integridad sobre los elementos de los conjuntos: tutor y estudiante. En general, cualquier información a-

cerca de la semántica de los datos, constituye una restricción de integridad, por lo tanto, se puede expresar mediante reglas.

Llegados a este punto, estamos en disposición de definir formalmente las reglas.

Sea V el conjunto de variables entidades y llamemos EV a $E \cup V$ e IV a $I \cup V$, se define REGLA como el par $\langle L, M \rangle$, donde $L, M \subset (EV \times IV \times EV)$. Informalmente, se puede utilizar como forma general de una regla, a la siguiente expresión:

$$\begin{array}{c} \text{expr-1 de hechos} \Rightarrow \text{expr-2 de hechos} \\ (L \Rightarrow M) \end{array}$$

Otro concepto importante, antes de definir lo que se entiende por una base de datos no estructurada, es el concepto de CLAUSURA: Sea H un conjunto de hechos y R un conjunto de reglas, se llama CLAUSURA de H bajo R , al conjunto de hechos que se obtienen de aplicar repetidamente las reglas de R a los hechos en H . Toda clausura de H incluye a H .

Finalmente, podemos definir BASE DE DATOS NO ESTRUCTURADA como una base de datos constituida por un conjunto H de hechos y un conjunto R de reglas, tales que la clausura de H bajo R está libre de contradicciones ([4]).

3. DICCIONARIO DE DATOS PROPUESTO.

Siguiendo los conceptos planteados en la sección 2, el diccionario de datos que se propone para un Sistema de apoyo a la toma de decisiones, es una base de datos no estructurada.

El desarrollo del diccionario comienza con la etapa de análisis del sistema DSS. Durante esta primera etapa, al descubrir los datos y su semántica, se determinan las entidades, los atributos y las interrelaciones. Con estos elementos se construyen los hechos del diccionario.

En las interrelaciones, de alguna manera está implícita la semántica de los datos, sin embargo existen otros aspectos que es necesario hacer explícitos, como lo son las dependencias funcionales, las reglas de pertenencia de las entidades a las interrelaciones y las dependencias entre elementos de distintos conjuntos entidad. Esto se puede lograr definiendo reglas, como las expresadas en la sección 2.

Con referencia a las restricciones de integridad ([6]), éstas también deben hacerse explícitas en el diccionario, lo cual se lleva a cabo definiendo otras reglas similares a las semánticas.

El diccionario de datos para un DSS, además de ser no estructurado, tiene ciertas características asociadas a su naturaleza de diccionario ([1,8]). Estas características surgen de que él contiene información descriptiva de los elementos de una base de datos. De acuerdo con ello, se definen algunos hechos particulares, muy útiles en este caso. Estos hechos se aplican a los elementos de datos que se describen en el

ORIGEN describe, para un elemento de datos, cual es la fuente que lo provee (por ejemplo: una planilla, una persona, etc.).
 Notación: (elem-dato, ORIGEN , fuente)

ES-ATRIBUTO indica que un elemento de dato es atributo de una entidad o interrelación.

Notación: (elem-dato, ES-ATRIBUTO , e)
 donde e puede ser el nombre de una entidad o de una interrelación

ES-CLAVE indica que un elemento de datos es clave de acceso de una entidad o una interrelación.

Notación: (elem-dato, ES-CLAVE , e)

PERTENENCIA-I indica como es la pertenencia de las entidades a las interrelaciones ([6] pags. 70 y 71). Esta pertenencia puede ser de varios tipos:

Opcional: cuando la existencia de las dos entidades participantes no depende de la interrelación.

Contingencia: cuando la existencia de una de las dos entidades es dependiente de la interrelación.

Condicional: cuando la existencia de una de las entidades depende de la interrelación, de acuerdo a un criterio que se puede expresar con una condición booleana. Es un caso particular de la contingencia.

Obligatoria: cuando la existencia de ambas entidades es dependiente de la interrelación.

Notación: (entidad, "tipo-pertenencia" , interrelación)

SINONIMO especifica un sinónimo para un elemento de dato.

Notación: (elem-dato, SINONIMO , elem-dato-1)
 donde elem-dato-1 es un sinónimo de elem-dato

Un aspecto muy importante en la descripción de los elementos de datos, que además afecta la integridad, es la definición de los dominios de los atributos. El dominio de un atributo es su conjunto de valores posibles. Los dominios se pueden definir como conjuntos, y luego se especifica cuales atributos pertenecen a cada uno de ellos, a través de un hecho de pertenencia. Por lo tanto, en el diccionario se debe incluir una facilidad para definir conjuntos tanto por extensión como por comprensión.

Consultas.

Para hacer consultas al DD no estructurado, se propone un mecanismo constituido por tres porciones: interrogación, navegación y búsqueda por sinónimos. A continuación se describe cada una de ellas.

Interrogación.

El mecanismo de interrogación a un DD no estructurado, que se propone, utiliza variables y "templates", similares a los construidos para expresar las reglas. Estos dos elementos forman parte de las fórmulas bien formadas que se utilizan para construir las interrogaciones. Una fórmula bien formada está definida por lo siguiente:

1. Los "templates" son los únicos predicados, y cada predicado es una fórmula atómica.
2. Si A y B son fórmulas y X es una variable, entonces $(A \wedge B)$, $(A \vee B)$, $(\forall x) A$ y $(\exists x) A$ son fórmulas.
3. Una expresión es una fórmula si y solo si se puede demostrar que es una fórmula, en base a las cláusulas 1 y 2, anteriores.

En las expresiones $(\forall x) A$ y $(\exists x) A$, se dice que A es el alcance del cuantificador. Se dice que la ocurrencia de una variable X está ligada en una fórmula, si está dentro del alcance de $(\forall x)$ o de $(\exists x)$, o si es la variable de los cuantificadores. En cualquier otro caso, la ocurrencia de la variable es libre en la fórmula. Una tupla $(c_1, c_2, \dots, c_n)$ satisface una fórmula Q con variables libres $(x_1, x_2, \dots, x_n)$ si $Q(c_1, c_2, \dots, c_n)$ se cumple.

Se define un QUERY (interrogación) como una fórmula que se construye con predicados, operaciones de conjunción y disyunción y con los cuantificadores universal y existencial.

Sea $Q(x_1, x_2, \dots, x_n)$ una fórmula con $x_1, x_2, \dots, x_n$ como sus variables libres. A Q se le llama QUERY (interrogación) y su valor se denota por $\{Q\}$ y es el conjunto de todas las tuplas $(c_1, c_2, \dots, c_n)$ tales que $Q(c_1, c_2, \dots, c_n)$ se cumple. Una fórmula sin variables libres es una proposición. Este formalismo es la base sobre la cual se apoya el mecanismo de interrogación. Cualquier requerimiento de información se puede expresar a través de él.

Navegación.

Existen situaciones en las cuales el usuario puede no saber qué preguntar o cómo hacerlo. En este caso, se utiliza la técnica de "browsing" ([4]). Como veremos a continuación, esta técnica tiene dos modalidades: navegación propiamente dicha y "probing" (indagación). En la primera modalidad de este tipo de consulta, se supone que el usuario no conoce la estructura de la base de datos. Se examina la "vecindad" de una entidad en particular, se toma otra entidad de allí y se repite el proceso de estudiar su vecindad, hasta que el usuario encuentre lo que busca o decida no buscar mas. Esta modalidad utiliza el tipo de "query" mas sencillo, que es el "template" con una o dos variables libres: (x_1, r, x_2) . Además, se utiliza el símbolo "*" para indicar todos los nombres de variables libres.

La modalidad de "probing" de este tipo de consulta, permite que el usuario siga una estrategia de ensayo y error cuando conoce algo sobre la base de datos. El punto

crítico de esta modalidad se encuentra cuando el usuario da un "query", y recibe como respuesta el conjunto vacío, de lo cual, en general, no se obtiene mucha información. Esta modalidad entonces, se basa en las reglas de inferencia por generalización. El usuario inicia su búsqueda con un "query", si éste falla, se intenta con otros "queries" derivados del primero, aplicando las reglas de inferencia por generalización. Al conjunto de "queries" derivados de esta forma, se le llama "conjunto de retracción" de Q, se intenta con cada uno de los elementos de él y se le reportan los éxitos al usuario.

Búsqueda por sinónimos.

Cuando durante el desarrollo de una base de datos intervienen varios usuarios, cada uno de ellos puede tener una visión diferente de los datos y requerir información distinta a partir de ellos. Uno de los objetivos de construir una base de datos, es la integración de las diversas visiones, en una visión consolidada que satisfaga a todos los usuarios. Sin embargo, en este intento, pueden presentarse situaciones conflictivas donde no sea posible poner de acuerdo a los usuarios, en cuanto al nombre de entidades o de elementos de datos. En tales casos, es de gran ayuda, permitir que cada usuario siga llamando a los ítems por el nombre que considere adecuado, para ello se deben registrar en el DD los sinónimos de cada ítem, de acuerdo al hecho especial SINONIMO, descrito anteriormente en esta sección. Luego, a la hora de hacer consultas, el diccionario debe proveer una facilidad de búsqueda por todos los sinónimos. En particular, cuando un "query" falle, este método debe analizar todos los "queries" que se puedan obtener a partir del fallido, sustituyendo las entidades, los elementos de datos y las interrelaciones, por los sinónimos que estén definidos para cada uno de ellos, si es que los hay.

Construcción de la base de datos a partir del diccionario.

La utilidad del diccionario de datos en el desarrollo de la base de datos del DSS viene dada, principalmente, por dos aspectos: a) la descripción de los datos y su semántica; b) la construcción de la base de datos. Con lo descrito hasta el momento, se cubre el primer aspecto, ahora conviene definir qué facilidades ofrece el DD con referencia a la construcción de la base de datos.

El uso que se propone para el DD en este trabajo, es que sirva de "filtro" entre quien está diseñando el DSS y la base de datos de éste último. Esto se logra, generando a partir de él, una estructura para la base de datos. La estructura más adecuada para esta situación es la relacional, pues es la más cercana al modelo conceptual que sirvió de base al DD. En otras palabras, es la estructura más flexible de las tres mas conocidas (relacional, jerárquica y de redes) ([7]). De esta forma, la estructura se genera cuando el diseñador conozca suficiente sobre los datos como para ensayar una estructura con los datos reales. Luego de creada la base de datos, se realiza una carga real. Las consultas y actualizaciones posteriores se canalizan a través del DD no estructurado, el cual debe traducir dichos requerimientos, en operaciones sobre la base de datos. La

información referente a los metadatos, se consulta directamente en el diccionario.

Las labores de mantenimiento del DD, se llevan a cabo a través de una colección de operadores. Algunos de los operadores más importantes son: ESCRIBA, para agregar hechos al diccionario, a cada hecho se le asocia una clave de identificación; ELIMINE, para eliminar un hecho del diccionario, para especificarlo se puede escribir el hecho, o dar su clave de identificación; INCLUYA, para incluir reglas al diccionario, se le asocia una clave de identificación a la regla, y EXCLUYA, para eliminar una regla del diccionario, para especificarla se escribe o se da su clave de identificación.

Como complemento a los operadores, se incluye una facilidad de consulta al diccionario. Esta facilidad provee toda la meta información de la cual se disponga, para cada entidad y cada interrelación. Por ejemplo, para las entidades se puede consultar en cuáles hechos participa y cuáles son sus atributos; para cada interrelación se puede consultar cuál es su tipo, y como es la pertenencia de las entidades que asocia, a la misma.

A continuación se muestra un ejemplo muy simple, donde se ilustran algunas de las características descritas del diccionario propuesto. Suponga un ambiente académico, para el cual se desea desarrollar una base de datos, para tal fin se definen como entidades: profesor, con atributos nombre, cédula, categoría, teléfono y oficina; estudiante, con atributos nombre-e, cedula-e, n°carnet, carrera; y tesis con los atributos código y título. Entre estas entidades se establecen tres interrelaciones: TUTOR (profesor y estudiante), DIRIGE (profesor y tesis) y DESARROLLA (estudiante y tesis). Para definir esta base de datos en el diccionario propuesto se definirían conjuntos, hechos y reglas. En primer lugar se definirían los conjuntos ENTIDADES e INTERRELACIONES, con los elementos (profesor, estudiante, tesis) y (TUTOR, DIRIGE, DESARROLLA), respectivamente. Entre los hechos se definirían: (nombre, ES-ATRIBUTO, profesor), (cédula, ES-ATRIBUTO, profesor), (categoría, ES-ATRIBUTO, profesor), (teléfono, ES-ATRIBUTO, profesor), (nombre-e, ES-ATRIBUTO, estudiante), (cedula-e, ES-ATRIBUTO, estudiante), (n°carnet, ES-ATRIBUTO, estudiante), (carrera, ES-ATRIBUTO, estudiante), (código, ES-ATRIBUTO, tesis) y (título, ES-ATRIBUTO, tesis) para definir los atributos de cada entidad; (profesor, DIRIGE, tesis), (estudiante, DESARROLLA, tesis) y (profesor, TUTOR, estudiante) para definir las entidades que asocia cada interrelación; tipo(DIRIGE)=1:n, tipo(DESARROLLA)=n:1 y tipo(TUTOR)=1:n para definir los tipos de las interrelaciones; y (tesis, contingencia, DIRIGE), (profesor, opcional, DIRIGE), (estudiante, opcional, DESARROLLA), (tesis, contingencia, DESARROLLA), (profesor, opcional, TUTOR) y (estudiante, opcional, TUTOR) para definir los tipos de pertenencia a las interrelaciones. Seguidamente se muestran dos de las reglas que cabe definir para este caso:

1. $(X, \in, \text{estudiante}) \text{ y } (T, \in, \text{tesis}) \text{ y } (X, \text{DESARROLLA}, T) \text{ y } (\exists w) (w, \in, \text{estudiante})$
 $\text{ y } (w, \text{DESARROLLA}, T) \Rightarrow (w, =, X)$
2. $(T, \in, \text{tesis}) \Rightarrow (\exists w) (w, \in, \text{estudiante}) \text{ y } (w, \text{DESARROLLA}, T) \text{ y }$

$(\exists x) (x, \in, \text{profesor}) \text{ y } (x, \text{DIRIGE}, T)$

Nota: esta regla expresa la situación de que la pertenencia de tesis a las interrelaciones DIRIGE y DESARROLLA, es de contingencia.

4. CONCLUSIONES.

Actualmente se está trabajando en la implantación del diccionario de datos propuesto. En esta labor se está poniendo especial énfasis en los aspectos:

1. Búsqueda de una estructura de datos adecuada para almacenar los hechos y las reglas del diccionario. Tal estructura debe proveer accesos rápidos.
2. Lenguaje del diccionario. Este lenguaje debe ser amistoso ("user friendly") y cómodo, tanto en la definición como en la consulta y el mantenimiento del diccionario.
3. Interfase con la base de datos del DSS. En este aspecto deben considerarse dos puntos: a) cómo generar una estructura relacional para la base de datos del DSS, a partir de la meta información contenida en el diccionario de datos del mismo, y b) cómo traducir requerimientos expresados de acuerdo a los mecanismos de consulta del diccionario, en operaciones sobre la base de datos que respondan efectivamente a las preguntas y como efectuar esas operaciones.
4. Interfase con un manejador de base de datos (DBMS, data base management system). Este aspecto involucra elegir uno (o más) DBMS desde el cual poder utilizar las facilidades del diccionario de datos propuesto. Esto, por supuesto, tendrá un gran impacto sobre el aspecto 3, mencionado anteriormente.

El servicio que preste el diccionario de datos al sistema de apoyo a la toma de decisiones, se podría ampliar incluyendo facilidades de análisis y diseño de sistemas de información, con herramientas gráficas.

BIBLIOGRAFIA.

- [1] ALLEN, F.W.; LOOMIS, S. y MANNINO, M. "The integrated Dictionary/Directory System". Computing Surveys ACM, Vol. 14, N° 2, Junio 1982.
- [2] CHEN, P. "The Entity-Relationship Approach to Logical Data Base Design" Q.E.D. Monograph Series on Data Base Management, N° 6, Mass. 1977.
- [3] CURTICE, R.M. "DATA DICTIONARIES: An assessment of current practice and problems" IEEE 1981.
- [4] MOTRO, A. "Browsing in a loosely structured Database" Communications ACM, January 1984.
- [5] PETERS, L.J. "Software Design: Methods and Techniques" Yourdon Press N.Y. 1981.
- [6] TEOREY, T.J. y FRY, J.P. "Design of Database structures" Prentice-Hall N.J. 1982.
- [7] ULLMAN, J.D. "Principles of Database Systems" Computer Science Press - 1982.
- [8] VAN DUYN, J. "Developing a Data Dictionary System" Prentice-Hall - N.J. 1982.