Generative Graphs and Representation by Induction of Orders: the RIO General Model of EURECAS

Roger Attias*

Université Paris 7, Institut de Topologie et de Dynamique des Systèmes, CNRS URA 34, 1 Rue Guy de La Brosse, 75005 Paris, France

Received September 9, 1992

A general model is proposed for the representation of topological information in the domains where objects are expressed by graphs. It formalizes the data organization and computer representation that I have earlier designed for substructure storage and retrieval in very large databases. Concepts are introduced at three successive levels of abstraction. The generative graph is an abstract class of isomorphic generic graphs sharing identical data structures. A point of view is an ordered generic graph related to a generative graph; it selects and organizes different semantic contents which anticipate the variety of situations to be handled. By induction of its order, the specific graphs, belonging to the associated class, are homogeneously represented (aspects), and common features are thus explicited. This redundant storage, based on the notion of ordered graphs, allows simple pattern recognition and limit combinatorial processes. The RIO concepts lead to formally related classes (abstract or generic levels) which are hierarchically organized and which allow the inheritance of properties and structures; they contribute to the conceptual clarity of a domain perception and suggest efficient data structures and simple procedures.

INTRODUCTION

Knowledge representation contributes to the formal perception of a domain. The main effort has been devoted in the early[1] or in the subsequent[2] models to disciplines such as natural language, human thought, expert systems, ... which present complex problems to solve: exceptions, modification of the universe, implicit or uncertain or ambiguous data, ... On the contrary, chemistry is a highly formalized domain within the context, however, of some restricted representation of the molecule. Structural chemistry and its mathematical tools provide an environment which allows abstraction and rigorous classifications.

A formal model, based on a previous work,[3a] is proposed for a multipurpose representation of topological information. It is a generalization, and the exact model, of the generic concepts and the data structures that I have earlier designed for substructure searching[3b] and implemented in the CBAC CAS File and then CAS Registry Structure File (EURECAS)[4,5] within the context of the DARC project. Actually, these chemical applications on large files constitute, for the main aspects, a validation of the model.

"Knowledge includes abstractions and generalizations of voluminous material":[6] in order to handle very large structural files and to exploit the properties of the topological information, we introduce three levels of abstraction with the notions of generative graph, points of view, and aspects.[3a,5] Algorithmically defined hierarchies are constructed with property and data structures inheritance. Semantic aspects of a graph are explicitly represented by appropriate ordering of the graph, emphasizing each, generic or specific selected features.

The Example of Chemistry: Evolution of Structural Representation in Retrieval Systems. Structural representations in chemistry are linked to retrieval systems. With the historical evolution of technology and derived techniques, they could progressively include a larger number of parameters, with increasing complexity, and handle new data types. The important development of the initial fragment codes is explained by their straightforward use by mecanographical equipment; then, linear notations were adapted to alphanumerical input/output of the early computers. Improved topological and geometrical representations have been more widely used only recently, owing to graphics capabilities and high speed computers.

Around the 1970's an important research effort was devoted to substructure search systems, since this problem is involved in information retrieval as well as in computer-aided design strategies. These systems implicitly define the entities effectively used to handle molecules by computer, with no constraint of a priori chemical significance.

Substructure systems and substructure retrieval systems have been recently fully reviewed.[7] We classify retrieval systems in three broad classes:

(1) Systems based on information science general techniques, where the presence or absence of some predefined part of the molecule is searched for. The techniques which are involved are to some extent an adaptation of textual processing. The most typical are nomenclature or fragment systems. Further developments, based on connectivity matrices, have resulted in more complex screening systems, which handle generic aspects or include hierarchized open sets of screens, but at some step, the previous principle holds: recognition of a predefined entity extracted from the query and then global matching with the preprocessed stored data.

(2) Systems based on technology (e.g., the CAS "search machine"). The preceding techniques are used together with parallel processing and networking. The initial problem is, thus, considerably modified by a new perception of processing time and of the volumes.

* Present address: Université René Descartes, Laboratoire de Chimie et Biochimie Pharmacologiques et Toxicologiques, CNRS URA 400, 45 Rue des Saints Pères, 75006 Paris, France.

0095-2338/93/1633-0415$04.00/0

(3) Systems based on the representation of structural information by a formal model. The information is logically and physically progressively organized, with redundancy: the exact generic features of the query are processed. The data organization, which has been implemented in EURECAS, is an early[3a] example; it is formalized by the general model presented below: the RIO (represention by induction of orders) model.

BASIC CONCEPTS

We consider a domain where objects are modeled by graphs $G_0(X, U, Fx, Fu)$ where Fx is a mapping of the set X of vertices onto a set Ex, and Fu is a mapping of the set U of edges onto a set Eu.

In chemical applications, Ex is the periodic table of elements and Eu is the set of bond values.

We call generic graphs of order 1, or simply generic graphs,[8] the graphs $G_1(X, U, Fx, Fu, R)$ where:

Fx is a mapping of X onto $P^*(Ex)$

Fu is a mapping of U onto $P^*(Eu)$

R is a mapping of X onto the set N of integers

$P^*(Ex)$ and $P^*(Eu)$ are respectively the sets of parts of Ex and Eu where the null element is excluded.

R will be called *residual adjacenty function*: it associates to every vertex the number of additional edges which are allowed to be incident to this vertex; this number is the *residual adjacency* of the vertex.

In structural chemistry, the entities modeled by such graphs are called generic substructures (or query structures in structural retrieval). They implicitly define substructures with three types of indeterminations:[4] list of values assigned to a vertex, list of values assigned to an edge, and range of connectivity degrees on a vertex.

An important property of this reductionist model is that it lends itself to simple mathematical handling while its expressive power is sufficient enough for most CAD problems.

Generic structures are expressed by G_1 graphs where R is null for every vertex.

Specific substructures are defined as entities expressed by G_0 graphs associated with an R function with a null value for internal nodes, when they can be defined.

TOPOLOGICAL RELATIONSHIPS

Substructure and Specific Structure. Consider a specific graph $G_0(X_0, U_0)$ and a generic graph $G_1(X_1, U_1)$. G_1 is said to be a *Substructure of* G_0, if there is a mapping f between G_1 and a partial subgraph of G_0, compatible with vertex values, edge values, and connectivity degrees, i.e.:

$$\forall x_1 \in X_1 \text{ and } \forall u_1 \in U_1 \text{ with } x_0 = f(x_1) \text{ and } u_0 = f(u_1)$$
$$Fx(x_0) \subseteq Fx(x_1)$$
$$Fu(u_0) \subseteq Fu(u_1)$$
$$\text{DEG}(x_1) \leq \text{DEG}(x_0) \leq \text{DEG}(x_1) + R(x_1)$$
$$\text{DEG}(i) \text{ being the degree of } i$$

This relationship will be written: G_1 sST G_0. Conversely G_0 will be called the *specific structure of* G_1, which is noted: G_0 SP G_1.

We emphasize the consequence of the connectivity constraint in the definition of the substructure relationship: a subgraph of G_0 is not necessarily a substructure of G_0 (e.g., considering the graph structure representation in chemistry: isopropyl is a substructure of isobutane, while propane is a subgraph but not a substructure of isobutane). This distinction allows easy definitions of disjoint classes or formal handling of exact/embedded matching.

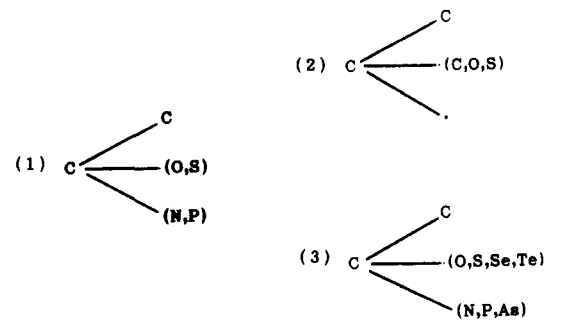

Figure 1. Comparing generic graphs. Graph 1 is a restricted substructure of graphs 2 and 3.

Restricted and Extended Substructures. The preceding relationships compare a specific graph G_0 with a generic graph G_1. We extend them in order to compare two generic graphs $H_1(X_1, U_1, R_1)$ and $H_2(X_2, U_2, R_2)$. H_1 will be called *restricted substructure* of H_2 (Figure 1) if H_1 includes a partial subgraph which is isomorphic to H_2 according to a function f such as:

$$\forall x_1 \in X_1 \text{ and } \forall u_1 \in U_1 \text{ with } x_2 = f(x_1) \text{ and } u_2 = f(u_1)$$
$$Fx(x_1) \subseteq Fx(x_2)$$
$$Fu(u_1) \subseteq Fu(u_2)$$
$$\text{DEG}(x_1) \geq \text{DEG}(x_2)$$
$$\text{DEG}(x_1) + R_1(x_1) \leq \text{DEG}(x_2) + R_2(x_2)$$

This relationship will be written H_1 RsS H_2. It implies that if H_1 is the substructure of a specific graph S, then H_2 is also a substructure of S:

$$H_1 \text{ RsS } H_2 \text{ and } H_1 \text{ sST } S \rightarrow H_2 \text{ sST } S$$

Conversely, H_2 will be called the *extended substructure* of H_1, which will be written H_2 XTs H_1.

In chemistry, the genericity levels of substructures having certain common features can be thus expressed and compared.

GENERATIVE GRAPHS

We call graphs of order 2 or *generative graphs* the graphs $G_2(X, U, Fx, Fu, R)$ where:

Fx is a mapping of X onto $PP^*(Ex)$

Fu is a mapping of U onto $PP^*(Eu)$

R is a mapping of X onto the set N of integers

$PP^*(Ex)$ and $PP^*(Eu)$ are respectively the sets of parts of the sets of parts of Ex and Eu where the null element is excluded.

Practically, in a generative graph, *lists of lists* of values are assigned to the edges and to the vertices.

Abstract Class. The generative graph is an *abstract class* of isomorphic generic graphs, such as each edge or vertex value in the generic graph is an element of the list of lists assigned to the corresponding edge or vertex in G_2 (Figure 2).

The set of elements of this class is exhaustively generated by a simple combinatorial process. It will be noted $\bar{G}_2$.

G_2 and the elements of $\bar{G}_2$ being isomorphic, a fundamental property of the concept of generative graph can then be stated: all the elements of an abstract class G_2 share common data structures and common properties. Defining an order on G_2 induces an order on the generic graphs, elements of its class, and, as a further consequence, on the elements of $\bar{G}_2$ (see below).

A generic graph being in turn a class of specific graphs, every such class inherits properties (structures, methods) from its abstract class.

Relationships between Generative Graphs. The preceding RsS and XTs relationships can further be extended in order to compare generative graphs. Without detailing them, we define, therefore, similarly two types of relationships based

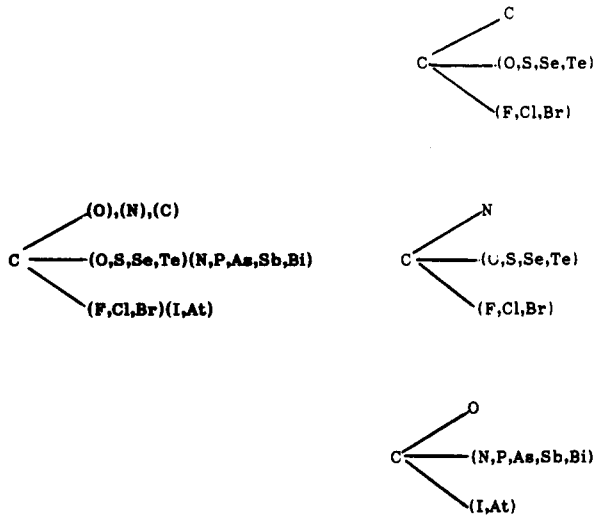

Figure 2. Generative graph and some of the isomorphic generic graphs of its abstract class. The isomorphism defines implicitly a common order labeling.

on the recursive comparison of lists of lists rather than on single lists: Extended Generative Graph (denoted XGG) and Restricted Generative Graph (denoted RGG). If P_1 and P_2 are two generative graphs such that P_1 XGG P_2, then $\bar{P}_2 \subseteq \bar{P}_1$ and a generic graph S_{1i} of P_1 is an extended substructure of the related generic graph S_{2i} of P_2: S_{1i} XTs S_{2i}.

Stereotypical Classes. The previous formal classification is the type of theoretical goal to be ideally reached by knowledge representation in most disciplines. In domains, like structural chemistry, where objects can be topologically represented, this goal can be simply achieved; the concept of a generative graph provides the tools for abstraction and sets a rigorous framework for the representation of the domain.

The graph isomorphism process, underlying the construction of $\bar{G}_2$, generates *stereotypical classes* which lead, as a practical consequence, to their homogeneous simple computer representation (fixed format, general parametrized procedures). This aspect is highly important when designing operational systems or when processing large files.

Each application will be characterized by an *appropriate choice* of generative graphs; they represent synthetically (mathematically) the pieces of semantic information selected to handle the particular situations occurring in the field under consideration.

Generic Extension. We call *generic extension* of a specific graph G_0 according to a generative graph G_2 the subset of elements of $\bar{G}_2$ which are substructures of G_0 (Figure 3); the corresponding operator will be denoted G_2*:

$$G_2^*(G_0) = \{G_{1,1}; G_{1,2}; \ldots; G_{1,n}\}$$

with $G_{1,i}$ sSt G_0 and $G_{1,i} \in \bar{G}_2$.

G_2* is the grammar which recognizes all the substructures of G_0, under a syntactic constraint.

A generative graph G_2 is a super class of G_0 graphs subdivided into G_1 subclasses. While *generating* all the valid substructures of a specific graph G_0, G_2* *assigns* G_0 to each of the G_1 classes to which it belongs. When applied to a family F of graphs, each element of F is assigned to 0 (no substructure match), 1, or several elements of G_2; the classification of the elements of F, according to common features which belong to a preselected, potentially large set, is thus *algorithmically* obtained.

These operations of recognition and extraction of generic or specific parts of objects are essential in the different fields of information processing.

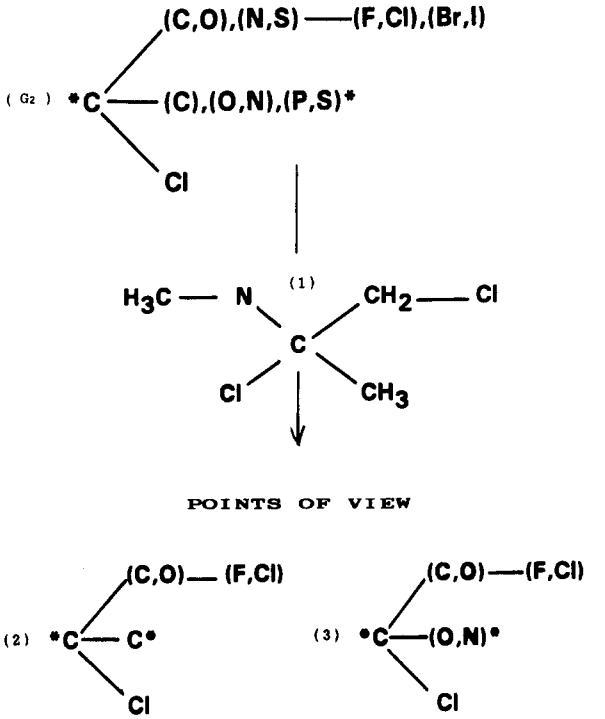

Figure 3. Generic extension of the graph 1 according to the generative graph G_2 is the set of graphs {(2),(3)} which are substructures of 1 and belong to $\bar{G}_2$.

CONSTRUCTION OF HIERARCHIES

We use the preceding concepts as the primitives of a semantic model where entities are progressively organized from generic to specific classes and at varying levels of abstraction. The classes of the successive levels are related by sST, XTS, or XGG relationships which infer inheritance of topological properties.

A symbolic root consists of a single vertex x_0 to which is assigned the value M_R (highest R value) and the value M_X (Fx value corresponding to the highest genericity).

A graph is described progressively by constructing the levels of the hierarchy, with increasing specificity, under the following semantic integrity constraint: a node j is the son of a node i only if $G_i(X_i, U_i)$ (the graph assigned to i) is a partial subgraph of $G_j(X_j, U_j)$ (the graph assigned to j), compatible with the node and edge values, the degrees, and the residual adjacencies. This is expressed for generic graphs by G_i XTs G_j.

The associated relationship $R(G_i, G_j)$, being reflexive, transitive, and antisymmetrical, is a partial-order relationship.

A node is a specialization of its father (restriction of genericity); brothers are subclasses of the father class from which they derive by applying a combination of simple operations expressing the XTs relationships: restriction of vertex, edge, or residual adjacency values or adjunction of edges and related vertices.

The design of specific hierarchies implies the choice of application-oriented constraints: e.g., a set of subclasses may be assigned the constraint of sharing a common graph skeleton, which is globally expressed by a generative graph.

A node of the hierarchy is the result of a valid transformation of its predecessor. We define here four transformation primitives; a valid transformation is any combination of these canonical operators:

Canonical Operators.

(a) Restriction of the residual adjacency of a vertex, with a label k from a value v to a value v' ($v' < v$): RESTR – $R(k, v')$. In chemical applications, this operator limits the number of additional substitutions.

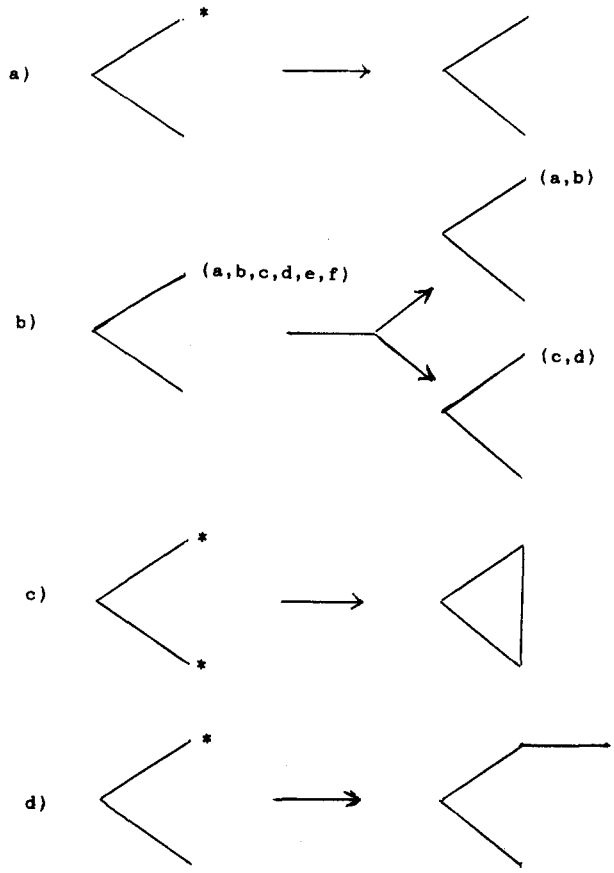

Figure 4. Four canonical operators. Each type of transformation is schematically represented. Nonnull residual adjacency values are indicated by an asterisk.

(b) Restriction of the value v_0 assigned to a vertex or an edge, labeled k, to a new value v_1 with $v_1 \subset v_0$: RESTR $- V(k, v_1)$.

(c) Adjunction of an edge with a label a between two existing vertices k_1 and k_2: EDGE(k_1, k_2, a). This operator decrements by 1 the values of the residual adjacencies of k_1 and k_2; it is then applicable only if they are not null, i.e., $R(k_1)*R(k_2) \neq 0$. This operator is not applicable for trees.

(d) Adjunction of an edge e and its related vertex v to an existing vertex k: NODE(k, e, v). The condition to be satisfied by the vertex k is $R(k) > 0$. After the operation, $R(k) \leftarrow R(k) - 1$. The implicit values for e and v, newly created, are the standard maximum values: $Fx(v) = M_X$; $Fu(e) = M_U$; and $R(e) = M_R$.

These operations are illustrated (Figure 4), and a simple hierarchy is shown (Figure 5) in the field of chemistry; it includes examples of disjoint classes (first level) and overlapping classes (second level).

In a hierarchy, a valid graph assigned to a vertex v is the transformation of its predecessor by the application of a set of canonical operators. Consequently, if we associate formally this set of operators to each edge of the hierarchy, the graph assigned to v is constructed by applying to the root the successive operators of the path (ROOT, v).

A generative graph being the generative grammar of generic graphs, a hierarchy of generative graphs induces a hierarchy over generic graphs.

With no loss of generality, and for the sake of clarity, the graphs considered in the following are trees.

POINTS OF VIEW

Let G_2 be a generative graph, G_0 a specific graph, and G_1 a generic graph element of $G_2^*(G_0)$, i.e., G_1 sST G_0.

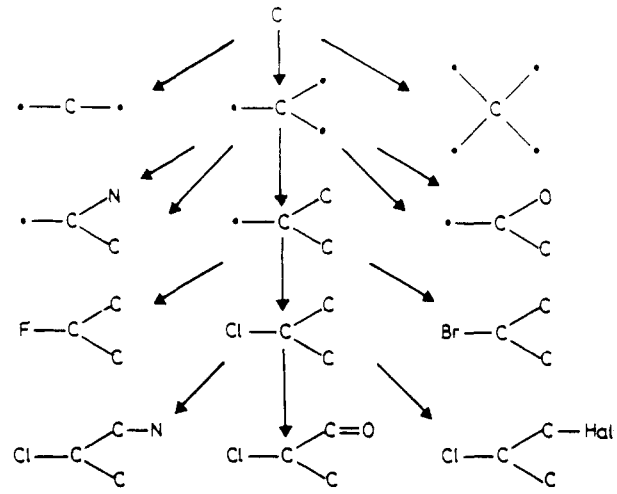

Figure 5. RIO concepts lead to different types of hierarchies. Example of hierarchy of generic graphs. A node is an extended substructure of its sons. Brothers may share a common graph skeleton and belong to the abstract class of a generative graph.

Let L be a process which assigns an ordering label to G_2 and, therefore, to G_1. We call *point of view of* G_0 *according to* G_2 the resulting labeled graph G_{1L}. It exhibits a choice of features of G_0 *intentionally* expressed by G_2. This notion is to be used together with the concept of induced order and the concept of aspect, which are introduced in the next paragraph.

The labeling process L can be any process; it provides an application-oriented canonical order, globally defined for a class of generic graphs by induction of the order of the corresponding generative graph (see below).

Alternatively, the point of view may also be intrinsically defined by ordering a generic graph as a whole, independent from a generative graph. We propose a conventional process L_0 for ordering generic trees as follows:

e_i is the incident edge to vertex i. Associate to i the list l_i of the couples c_{ij}, elements of the product set Ex^*Eu, which combine the values $Fu(e_i)$ assigned to e_i with the values $Fx(i)$ assigned to i: $l_i = \{c_{ij}\}$ with $c_{ij} = \{u_{ik}, x_{il}\}$ and $u_{ik} \in Fu(e_i)$, $x_{il} \in Fx(i)$.

The vertices are ordered recursively level by level. The vertices i sharing the same predecessor (set of brothers) are ordered according to the successive following criteria of priority: shortest list li, highest value u_{ij}, and highest value x_{ij}.

Other conventional orders may be chosen in order to exploit, for instance, the particular situations which occur in the field under investigation.

INDUCED ORDERS

Induction of Order by a Point of View. G_1 and G_{1L} are isomorphic to a partial subgraph of G_0. We use the 1:1 correspondence to assign the labeling of G_{1L} to this partial subgraph. We call *induced order* of the point of view this resulting order on G_0. It differs from canonical numbering of specific graphs since it is not based on the whole graph but only on selected generic or specific features called point of view.

Induction of Order by a Generative Graph. We define an order on generative graphs by extending the rules applied to generic graphs in the previous paragraph and by considering lists of lists of values instead of lists of single values. A similar lexicographical ordering of the sets of values assigned to the vertices is obtained and priority labels are inferred.[3a]

The resulting order, at the generative graph level, induces an order labeling on each of the generic graphs belonging to

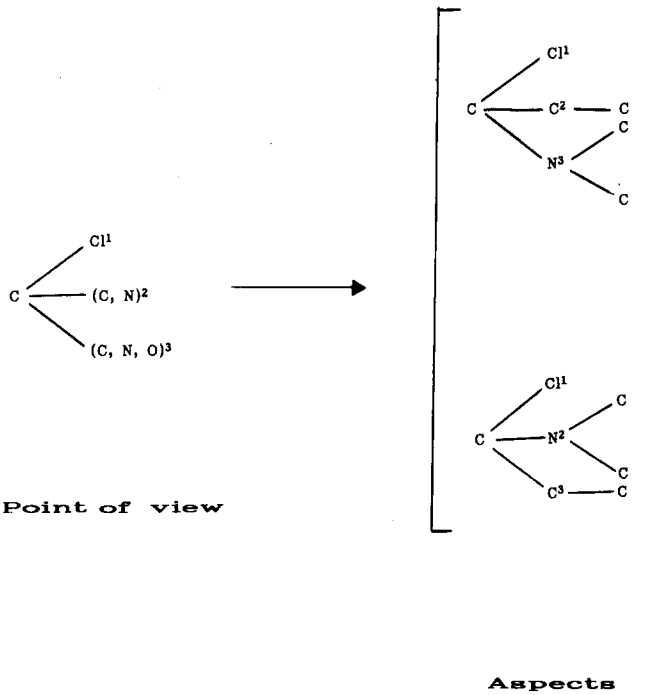

Figure 6. Two *aspects* represented are two different ordering of the same graph which are both compatible with the *point of view*. This multiple interpretation of a graph allows efficient processing based on the notion of order.

the class by a straightforward process: the 1:1 correspondence between the generative graph and each of the related isomorphic generic graphs, is used also for the label correspondence (Figure 2).

This canonical global labeling of a set of generic entities depends solely on their membership class, and reflects specific semantic and structural properties; it provides its full expressive power to the representation model.

As a simple illustrative example, the list l_i of lists l_{ij} of values assigned to the vertex of a generative graph may correspond to the sets of alternative generic possibilities for a given parameter. The common labeling sets an explicit correspondence between similar vertices (each being assigned a different l_{ij} list in each subclass of the abstract class), and contributes to an enhanced representation by logically increasing its semantic power. It has important practical consequences: this additional information leads to simple data structures, the identification and the access to selected aspects of the graph is easy, and procedures lend themselves to parametrization.

In chemistry, l_i is assigned to a single site, and the different l_{ij} lists are the sets of atoms expressing the specific atom grouping selected for this site in a given application.

ASPECTS OF A GRAPH

A point of view G_{1L} of G_0 is isomorphic to one or several partial subgraphs of G_0. It induces an order on each of these graphs. We call *aspects* of G_0 according to G_{1L} these ordered graphs which are elements of the isomorphism class of G_0. The multiplicity of isomorphisms is shown in Figure 6 where two aspects of G_0, corresponding to two different labeling, are compatible with the same point of view.

All the *instances* of the ordered substructure (point of view) which occur in a graph are thus exhaustively identified. This preprocessing is ultimately compensated by computational efficacy; the assigned order is an explicit additional semantic information which selects and organizes common patterns, avoiding further combinatorial processes. The multiplicity

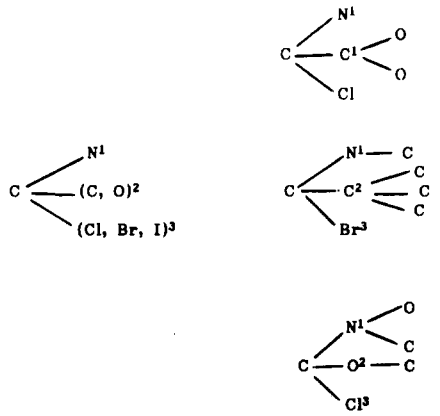

Figure 7. Aspects of a class are labeled by *induction of the order* of the related point of view. The common substructure is thus exactly identified; its unique representation is independent from the varied structural contexts.

of representation anticipates also the diversity of situations, within varied contexts, which arise in a specific field.

The point of view being associated to a generative graph, the aspects reflect also the ordering and the properties of this generative graph.

Enhanced Classification. A point of view represents the class of ordered graphs sharing common features, i.e., admitting a common substructure. The representation of these graphs (aspects) does not depend on the structural context of occurrence of their common substructure but on the common canonical order induced by the point of view (Figure 7). Sharing this canonical order has an important consequence: all the graphs in a given class are homogeneously stored in semantically significant data structures, allowing efficient processing.

Initially a generic graph was perceived as a class of specific graphs. This notion can now be refined by considering ordered graphs, i.e., the *isomorphism class* of each graph. A point of view, which is an ordered generic graph, is a class of aspects (ordered specific graphs).

This classification involves meaningful data *redundancy*, since several aspects of the same graph may contribute to a class, each instance being a different interpretation of the generic features of the class.

In each application (e.g., CAD in chemistry), this type of abstraction contributes to an improved analysis of the field.

Semantic Contents. The aspects related to a given point of view each enclose different specific contents, but they express by their common induced labeling the same general semantic contents resulting from this particular "fuzzy view" of the graph (selection and omission of features, generalization of others, creation of a relative meaning between them).

A graph, or one of its subgraphs, may belong to more than one class; it is perceived differently from one class to another (i.e., from different points of view). The graph is interpreted and represented, in each class, as the specific structure of different generic graphs.

An illustrative example (Figure 8) in the chemical field shows a specific structure and two classes to which it belongs, with two different semantic perceptions. For instance, in class 1, the chlorine is seen as a specific value of an undetermined atom, and in class 2, the chlorine is interpreted as a specific halogen.

The generative graph generates homogeneous classes of generic graphs; the induced orders selected explicit semantic relationships among them. These concepts are the semantic complement to the generic/specific hierarchy and to the field perception and organization. Useful storage and retrieval

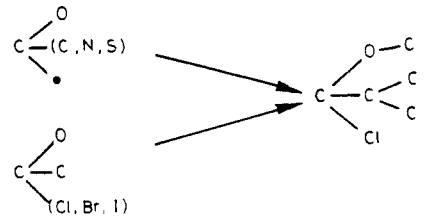

Figure 8. Two points of view of the same graph. They each express a different semantic interpretation of the graph.

techniques are derived; they will be further detailed for substructure searching in chemistry.

Extensions. In structural chemistry, an important research effort has been devoted to the handling of patents (Markush structures); they are a compact representation of a set of structures and may include, in addition to the structural diagram component (graph component), textual components, recursive generic definitions, and variable parameters. The description of the varied types of genericity and specificity has been achieved generally by the design of a single general language.[9-11] The notions of intensional and extensional representation of chemical entities has been fully discussed in this context.[12] In different systems, the analysis of the domain has resulted in the definition of disjoint generic classes and overlapping generic classes,[10] which are adapted to the goals and requirements of the field, of the users and of the processing strategies.

The concepts of generic and generative graphs, induced orders, and related hierarchies and relationships provide a framework for the representation and the study of this type of genericity. More specifically, the data organization, the data structures, and the retrieval mechanisms of the RIO model have been applied efficiently to the field of Markush structures (DARC).

In more general terms, the RIO model represents and organizes generic entities expressed by a generic graph as defined in the preceding sections. Other types of genericity may be handled; they require extended conventions and models: additional formal analysis of the domain, additional classifications, processing of nontopological information, ...

CONCLUSION

The RIO model is proposed as a general model for a semantic representation of information in the fields where objects are expressed by graphs. The tools it introduces are intended to provide a formal framework for reasoning and to suggest simple data structures and efficient retrieval procedures.

Sets of graphs are constructed according to redundant selected generic properties (overlapping classes of graphs sharing a substructure or an extended substructure) and to a multiple representation of basic entities (aspects, i.e., elements of the isomorphism class of a graph). They are formally interrelated by the generic/specific relationships between their levels of hierarchy and by the relationships between their levels of abstraction (e.g., membership to a generative graph).

The hierarchies are improved by introducing the notion of induced order, which is used to represent specifically different semantic contents of a graph. The multiplicity of these representations anticipates generic situations and limits, therefore, on line combinatorial processes; it contributes to *clarity* in the representation of the field under study and provides *classification* facilities.

The main basic generic concepts have originated in the restricted context of chemical substructure searching[3b,4] and result from a double objective:

Definition of a flexible structural language

On-line searching for *large* collections of structures, i.e., definition of a representation which is adaptable to efficient data structures

Our work was, therefore, first oriented toward the design of data structures in a retrieval system[3b] which could handle local isomorphisms, according to the generic structure types (infra-FRELs, query structures) that I had defined (intuitively at that time). The graph theory formulation of the generic entities, and of their relationships, is the basis of our initial model;[3a] it was intended to provide a synthetic presentation of the retrieval system and a general framework for subsequent applications in CAD in chemistry and in fields other than chemistry. The data representation which is implemented in the substructure search system remains a rigorous application of the RIO general model, the specific choices being expressed by an appropriate set of generative graphs.[5]

In subsequent papers, we will present the specific application of structural retrieval in chemistry. The generalized RIO model will then be further discussed and exemplified by an effective implementation[13] of the concepts on a very large data base (EURECAS).

REFERENCES AND NOTES

(1) (a) *Semantic Information Processing*; Minsky, M., Ed.; MIT Press: Cambridge, MA, 1968. (b) *The Psychology of Computer Vision*; Winston, P., Ed.; McGraw Hill: New York, 1975.
(2) *The Knowledge Frontier*; Cerone, N., Mc Calla, G., Eds.; Springer Verlag: New York, 1987.
(3) (a) Attias, R. Internal report (introduces the formal concepts: generic and generative graphs, and their induced orders), June 1979, ITODYS, University of Paris, France, 18 pp. (b) Attias, R. Internal report (proposing a substructure search system: generic entities, hierarchical organization, data structures and retrieval mechanisms), 1973, LCOP, University of Paris, France.
(4) Attias, R. DARC Substructure Search System: A New Approach to Chemical Information. *J. Chem. Inf. Comput. Sci.* **1983**, *23*, 102–108.
(5) Attias, R. EURECAS/DARC. La Sous-structure en Chimie.Contribution à la Représentation de l'Information Structurale et Application à la Recherche dans de Très Grandes Bases de Données. Thèse de Doctorat d'Etat, Université Paris, France, 179 pp, May 1992.
(6) Smith, J. M.; Smith, D. C. P. Data Base Abstractions: Aggregation and Generalization. *ACM TODS* **1977**, *2*, 105–133.
(7) Attias, R. Substructure Systems and Structural Retrieval Systems. In *Encyclopedia of Library and Information Science*; Kent, A., Ed.; New York, 1992; Vol. 50 (13), pp 308–363.
(8) Attias, R.; Dubois, J. E. Substructure Systems: Concepts and Classifications. *J. Chem. Inf. Comput. Sci.* **1990**, *30*(1), 2–7.
(9) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 2. GENSAL. A Formal Language for the Description of Generic Chemical Structures *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151–161.
(10) (a) Fisanick, W. Requirements for a System for Storage and Search of Markush Structures. In *Computer Handling of Generic Chemical Structures*; Barnard, J. M., Ed.; Gower: Brookfield, VT, 1984; pp 106–129. (b) Fisanick, W. The Chemical Abstracts Service Generic Chemical (Markush) Structure Storage and Retrieval Capability. 1. Basic Concepts. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 145–154.
(11) Tokizane, S.; Monjoh, T.; Chihara, H. Computer Storage and Retrieval of Generic Chemical Structures Using Structure Attributes. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 177–187.
(12) Dethlefsen, W.; Lynch, M. F.; Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Barnard, J. M. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 233–253.
(13) Articles in preparation to be submitted by the author: The application of the RIO model to chemical substructure searching in very large databases and EURECAS (ref 5 also).