Overcoming the Limitations of a Connection Table Description: A Universal Representation of Chemical Species

Susanne Bauerschmidt† and Johann Gasteiger*

Computer-Chemie-Centrum, Institut für Organische Chemie, Universität Erlangen-Nürnberg, Nägelsbachstrasse 25, D-91052 Erlangen, Germany

Received January 24, 1997⊗

The following paper describes a new model for the representation of chemical structures in information processing. This representation overcomes the limitations of connection tables designed to only represent chemical structures with bonds localized between two atoms. The representation introduced is based on the separation of the σ- and π-electrons of bonds and the delocalization of electrons also across more than two atoms. It also allows the description of chemical compounds containing multicenter or coordinative bonds. The representation was implemented using object-oriented programming techniques. The important classes of the developed class library are introduced.

1. INTRODUCTION

The solution of chemical problems by computational methods asks for the fundamentally important decision on how to represent chemical structures. The diverse goals of the many different computer applications that handle chemical structures have led to the development of a variety of structure representations. These representations are specially tuned to optimize the efficiency of such applications. Semiempirical and *ab initio* programs optimize the three-dimensional geometry of a structure with the aim of finding low energy minima. Therefore, the geometry and symmetry of a structure has to be well-defined in the chosen representation. Databases of chemical structures have to store large numbers of compounds. In order to make such a database effective, the following requirements have to be met: Each compound must be stored only once in compact form, and the database should be amenable to fast structure retrieval. This asks for the representation to be unique, unambiguous, nonredundant, and easily comparable. Programs that treat chemical reactions such as in synthesis planning or reaction prediction require a representation that allows the efficient treatment of the rearrangement of atoms and electrons in the course of a chemical reaction.

In databases of chemical structures and reactions as well as in synthesis planning systems a connection table has become the standard form for describing a molecule.[1] A connection table gives the list of atoms of a molecule and specifies the bonds between two atoms by listing the bond order. Connection tables have become so ubiquitous that formats like MDL's Molfile[2] are used as quasi standards for the exchange of molecular structure information. It was realized early on that a proper treatment of chemical reactions asks for the specification of the free electrons on atoms.[3]

Connection tables have the advantage that they are easy to use and interpret. However, it should be realized that a connection table is basically a description of a single valence bond structure. Thus, in a connection table the bond orders implicitly code electrons: Each single bond contains two electrons; in multiple bonds the number of electrons is twice the bond order. Compounds that are insufficiently described by such a fixed distribution of electrons in free electron pairs and diatomic bonds cannot properly be represented by a connection table. Figure 1 shows examples for such compounds. In the two bridging bonds of diborane (Figure 1a) two electrons are shared by three atoms; this type of bond cannot be coded in a usual connection table. Organometallic compounds like ferrocene (Figure 1b) contain bonds with bonding electrons being distributed between the metal atom and several atoms (here five) of the organic ligand. A representation by localized bonds such as required in a connection table is insufficient. The representation of organic species is often ambiguous when orthogonal π-systems have to be taken into account. In methoxyethene (Figure 1c) one of the lone pairs of the oxygen atom is in conjugation to the double bond; the other is orthogonal to it. A representation that only counts the free electrons on an atom cannot make such a distinction. Carbenes (Figure 1d) exist in two different states, as singlet carbene with one electron pair and one empty orbital at the carbon atom, or as triplet carbene with two unpaired electrons. The two states are not distinguishable by only counting the free electrons.

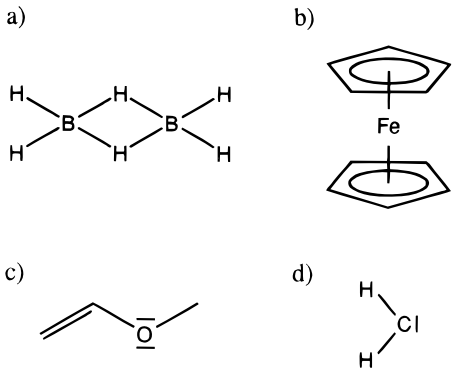

Figure 1. Compounds that cannot properly be described by a single valence bond structure and thus by a connection table: (a) diborane, (b) ferrocene, (c) methoxyethene, and (d) carbene.

Several approaches have tried to overcome the problems that arise from the incomplete representation that is inherent

* Author to whom correspondence should be addressed. E-mail address: Gasteiger@torvs.ccc.uni-erlangen.de.
† E-mail address: Bauerschmidt@torvs.ccc.uni-erlangen.de.
⊗ Abstract published in *Advance ACS Abstracts,* May 15, 1997.

S0095-2338(97)00442-3 CCC: $14.00

in a valence bond model and its equivalent of a connection table. Quite some time ago it was pointed out that the separation of σ- and π-electrons shows many advantages for the representation of structures that are better described by a molecular orbital model and also for the treatment of pericyclic reactions.[4,5] In the database systems designed by Chemical Abstracts Service and Beilstein two different solutions to deal with the problem of delocalization have been adopted. The main goal in their design of a structure representation was to ensure that the representation is unique and unambiguous. Each compound must always be represented by only one and the same connection table irrespective of initial atom numbering or mesomeric form of the structure. This is necessary in order to assure that only one database entry will be generated for each compound. Different solutions have been adopted by the two systems to overcome these problems.

In the connection tables used in the CAS system alternating double and single bonds are normalized into so-called alternating bonds.[6] The Beilstein registry system follows suggestions from ref 4. A connection table is built comprising all σ-bonds. Then, the valence electrons stemming from π-bonds are assigned to the atoms so that each atom is assigned one π-electron for each multiple bond it participates.[7]

In structure representation systems that are not designed for database storage but for usage in reaction prediction programs other prerequisites have to be fulfilled. Here the main goal is a representation that allows easy rearrangements of the structures. All valence electrons must be accounted for; no illegal valence states should be allowed.

Ugi et al. have extended their system of bond and electron matrices (BE matrices)[8] to be able to cope with delocalized σ- and π-bonds as well as coordination compounds. A BE matrix is a connectivity matrix that also codes the free electrons of each atom. In order to be able to describe delocalization, new bond types and extensions to the BE matrices have been introduced.[9,10] They were called *s*ymbolic *e*xtended *b*ond *e*lectron matrices (sXBE matrices). The bond orders are replaced by bond types that are not restricted to localized bonds. Localized bonds are represented by four bond types called single, double, triple, and quadruple bonds. The representation of delocalized bonds is split into two parts. First, three bond types are used to specify a delocalized bond between two atoms in the adjacency matrix. *Edsys* is used for electron deficient σ-bonds as in boranes, *pisys* for delocalized π-systems as in benzene, and *coord* for coordinative bonds in metal organic compounds. Furthermore, another row is added to the adjacency matrix containing nonzero entries for all atoms that are part of the same delocalized system together with the number of electrons contained in it. The main shortcoming of this approach is that for each pair of neighboring atoms only one bond type is allowed. One cannot describe compounds like benzyne or allene cations in which a delocalized and a localized bond are orthogonal to each other. In addition, different spin states like the S and T state of a carbene cannot be distinguished.

Dietz[11] uses a molecular multigraph, i.e., an undirected graph with vertices representing atoms and edges representing bonding relations. As two vertices in a multigraph may be connected by more than one edge, it is no longer possible to represent these structures in a simple matrix form. Instead, a pair of sets is used: The first set, the vertex set, comprises all atoms. Each of the atoms in turn is described by a triple containing the chemical element, the number of free valence electrons, and an index number. The second set, the edge set, is used to represent the bond information. A so-called bonding system is not restricted to connect only two atoms; it may extend over a larger number of atoms. Each of these bonding systems is described by the number of electrons contained in it and a set of atom pairs. All atom pairs together define the constitution of the compound, as the atoms of each pair are considered adjacent to each other. Therefore, there is no distinction between bond types. Regular single or multiple bonds are represented by a bonding system consisting of only two atoms, whereas a delocalized or coordination bonding system is formed by a larger number of atoms. However, this approach still treats all free electrons of one atom as one entity. Different spin and excited states cannot be modeled.

Our research group has developed several programs that deal with chemical reactions. EROS6.0[12] is a program system that predicts reactions of organic compounds. Different formal reaction types are applied to a set of starting materials. These formal reaction types describe how the valence electrons are rearranged in the course of a chemical reaction. Therefore, a proper description how the valence electrons are distributed in the starting materials is necessary.

WODCA[13] is a program system that offers several methods to assist in the design of organic syntheses plans. Powerful search methods can direct the user to favorable starting materials contained in compound catalogs. In addition, the development of retrosynthetic steps is necessary to explicitly derive the sequence of synthesis reactions between the starting materials and the target compound. Again, a detailed handling of the distribution of valence electrons is necessary.

MASSIMO[14] and FRANZ[15] are programs for the analysis and simulation of mass spectra of organic compounds. They model the fragmentations and rearrangements that take place in a mass spectrometer. Both generate a reaction network that consists of reactions of ionized organic compounds.

In these reaction networks, cations, radicals, and radical cations with delocalized π-systems have to be considered. Our first solution to this problem was to describe such species by an ensemble of all different valence bond structures. Although this process is quite laborious, we have developed an algorithm that performed quite well in most cases. However, it became clear that situations occur, where the description of such species like delocalized radical cations even by an ensemble of all valence bond structures is just inappropriate as the limitations of a connection table are met. This is the case with orthogonal π-systems that cannot be distinguished and where electrons or charges are shifted to atoms that are not part of the considered π-system. In order to overcome these difficulties and limitations we decided to change the basic structure representation from a valence bond based connection table to a new form where all these difficulties can be solved.

In addition, concepts are introduced that allow the treatment of species consisting of several ions and thus can represent electrostatic interactions in salts. Furthermore, the collection of several molecules into ensembles has particular advantages for the treatment of chemical reactions.

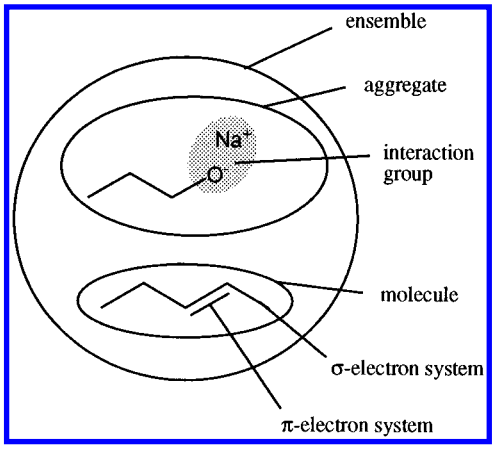

Figure 2. Composition of an ensemble.

In the following sections, first the major characteristics of the new representation are explained. Then, their implementation by object oriented programming techniques is described.

2. CHEMICAL STRUCTURES BEYOND A VALENCE BOND DESCRIPTION

As explained above, many chemical structures cannot sufficiently be described by a connection table that reflects only a single valence bond structure. The inherent characteristics of a connection table in specifying chemical bonds only by pairs of atoms and a bond order have therefore to be abandoned. In the approach described here, a basis for the description of different types of chemical bonds in a molecule is laid by the separation of σ- and π-electrons into two bond types, σ- and π-electron systems. A third bond type for coordination compounds is introduced.

First, the definitions used in the following discussion are given.

Electron System. Covalent bonds are described using electron systems. There are three types of electron systems: σ-electron systems, π-electron systems, and coordinative bonds.

Interaction Group. Noncovalent interactions between atoms or electron systems are treated by so-called interaction groups that contain the interacting atoms and electron systems and specify the kind of interaction between them. These interactions can be both intermolecular or intramolecular.

Molecule. A molecule is defined as a chemical entity where all atoms are bonded by covalent bonds. Not only neutral molecules but also ions or transition states are regarded as molecules in this context. The molecular graph that contains the connectivities of all atoms in a molecule is built using the σ-bonds and coordinative bonds.

A molecule is not the largest entity that can be described. Two different concepts of combining molecules have been introduced.

Aggregate. Aggregates are composed of a set of molecules or ions that are connected to each other by intermolecular interactions described as interaction groups.

Ensemble. Ensembles are composed of isolated molecules or aggregates. This feature allows one to combine species that are used in the same context (e.g., starting materials of a reaction).

Figure 2 illustrates the composition of an ensemble consisting of one aggregate and one molecule. The aggregate is composed of an ion pair that forms a salt. An interaction group is used to specify the electrostatic interaction between both ions of the aggregate. All atoms and electrons in each molecule are bonded by σ- and π-electron systems.

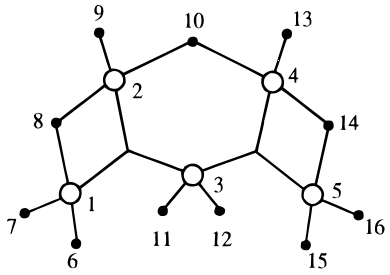

Figure 3. B_5H_{11} as an example for an inorganic compound with electron deficient multicenter bonds. The large empty circles denote boron; the small filled circles denote hydrogen atoms. The numbers next to each atom are the indices used for the specification of the electron systems.

A detailed description of these concepts with illustrating examples is given in the following paragraphs.

2.1. *σ*-Electron Systems. Multiple bonds are separated into the σ-part and the π-part of the bond. The electrons localized in the σ-part of the bond are represented by so called σ-electron systems. Single bonds are also described as σ-electron systems. Such σ-electron systems are not always simply single bonds, as also structures that contain electron deficient or electron excess multicenter bonds are treated here. Thus, σ-electron systems can also consist of more than two atoms when multicenter bonds are described. σ-Electron systems may contain a number of electrons different from two when ionized single bonds or electron excess compounds have to be described. Two types of multicenter σ-electron systems exist. The overlapping orbitals in *closed σ-electron systems* point into the central region between the collection of bonded atoms. In open *bridging σ-electron systems* one atom is located between the other atoms that are part of the σ-electron system. As examples for such multicenter bond types the closed BBB bond between atoms 1, 2, and 3 and the open bridging BHB bond between atoms 1, 2, and 8 in B_5H_{11} are given (see Figure 3). The following atoms and electron systems are necessary to describe B_5H_{11}:

atoms: B: 1−5; H: 6−16

σ-electron systems: (1 2 3 s c; 2) (1 6 s r; 2) (1 7 s r; 2) (1 2 8 s b 8; 2) (2 9 s r; 2) (2 4 10 s b 10; 2) (3 4 5 s c; 2) (3 11 s r; 2) (3 12 s r; 2) (4 5 14 s b 14; 2) (4 13 s r; 2) (5 15 s r; 2) (5 16 s r; 2)

The information in each parentheses describes one electron system. First, the indices of the atoms in the electron system are given, and then the type of system (s for σ, p for π, c for a coordinative bond) follows. The next entry marks the type of σ-electron system, r for a regular two-electron two-center bond, b for a bridging bond, and c for a closed bond. In bridging σ-bonds with a central atom this atom is then specified. At the end, the number of electrons is given, separated by a semicolon from the previous information.

Multicenter σ-electron systems occur not only in boranes but also in nonclassical cations and other organic reaction intermediates or transition states. A representative of an electron deficient σ-electron system is the 2-norbornyl cation

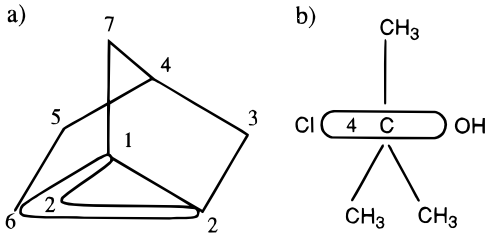

Figure 4. 2-Norbonyl cation (a) and the transition state of an S_N2 reaction (b) as examples for organic compounds with multicenter σ-electron systems. The regular σ-electron systems are drawn as lines; the multicenter electron systems are drawn as closed bent or oval shapes. The number of electrons is given in the shapes.

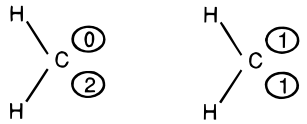

Figure 5. Free electrons of singlet and triplet carbene coded as π-electron systems. In the singlet carbene at the left-hand side both electrons occupy the same electron system; in the triplet carbene at the right-hand side each π-electron system is filled with one electron. σ-Electron systems are drawn as lines; π-electron systems are drawn as oval shapes.

(Figure 4a). The carbon atom framework can be described by the following electron systems:

(1 2 s r; 2) (1 2 6 s b 6; 2) (1 7 s r; 2) (2 3 s r; 2)
(3 4 s r; 2) (4 5 s r; 2) (4 7 s r; 2) (5 6 s r; 2)

The transition state of an S_N2 reaction can be modeled using an electron excess σ-electron system (Figure 4b). The electron pair that was the free electron pair of the attacking group as well as the electron pair that will become the free electron pair of the leaving group form this electron system consisting of four electrons. All other σ-electron systems are regular.

2.2. π-Electron Systems. π-Bonds and free electrons are both handled as π-electron systems. A π-electron system may consist of only one atom for the representation of free electrons or of several atoms for the representation of delocalized π-bonds. The number of electrons it may contain ranges between zero and twice the number of atoms.

Free electrons are coded as π-electron systems, which may contain at most two electrons. If an atom possesses more than two free electrons one π-electron system is generated for each electron pair. Thus, free electrons that often behave like π-bonds in reactions can be handled like electrons of other electron systems. In addition, different excited states of a molecule can now be distinguished. For instance, singlet and triplet carbenes can be described as shown in Figure 5. In a singlet carbene both electrons occupy the same electron system; in a triplet carbene two electron systems are each filled with one electron. These electron distributions are responsible for the different reactivities and properties of both states that can now be modeled.

If a molecule contains several orthogonal π-systems, one π-electron system is generated for each of them. The electrons of a triple bond, for instance, are distributed into one σ-electron system and two π-electron systems, each holding two electrons. This is important for the analysis and prediction of mass spectra, where it is necessary to show in which orbital ionization has taken place. Ionization of furan may take place either in the single lone pair on the

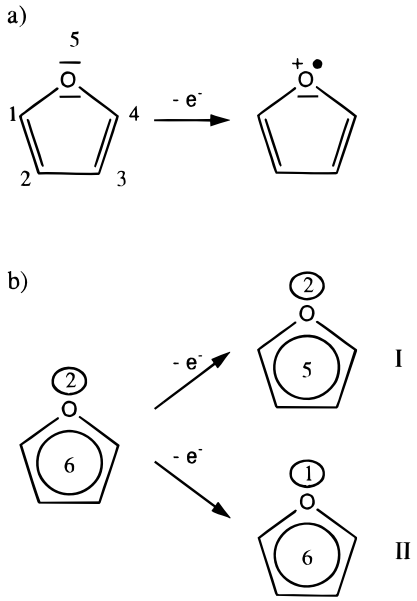

Figure 6. Ionization of furan: (a) ambiguous valence bond structure and (b) the representation using σ- and π-electron systems leads to two different structures for the ionization product.

oxygen atom or in the aromatic ring. A valence bond representation as illustrated in Figure 6a is ambiguous. It cannot be decided whether the charge and radical center are localized at the oxygen atom or delocalized in the aromatic ring system. The two different structures (Figure 6b) possible from ionization can be described when using the new description. The neutral compound is represented by the following atoms and electron systems.

atoms: C: 1−4; O: 5

σ-electron systems: (1 2 s r; 2) (1 5 s r; 2) (2 3 s r; 2)
(3 4 s r; 2) (4 5 s r; 2)

π-electron systems: (1 2 3 4 5 p; 6) (5 p; 2)

Ionization can give either (1 2 3 4 5 p; 5) (5 p; 2), I in Figure 6b, or (1 2 3 4 5 p; 6) (5 p; 1), II in Figure 6b, for the π-electron systems.

Consideration of the orthogonality of π-systems allows a more exact description for certain organic species, as will be shown in the example of Figure 7. One form of 1,2-diphenylacetylene is planar which corresponds to the conjugation of both phenyl rings through one π-bond of the triple bond. In another form each phenyl ring is conjugated with a different π-orbital of the triple bond. This corresponds to the geometry shown in Figure 7b, where both rings are perpendicular to each other. Using the notation of π-electron systems both forms can be represented separately. The planar form (Figure 7a) has one long and one short π-electron system

(1 2 3 4 5 6 7 8 9 10 11 12 13 14 p; 14) (7 8 p; 2)

The perpendicular form (Figure 7b) has two π-electron systems of equal size

(1 2 3 4 5 6 7 8 p; 8) (7 8 9 10 11 12 13 14 p; 8)

2.3. Coordinative Bonds. Coordination compounds may be formed by donation of electron density from electron rich ligands to a metal atom or cation. Coordinative bonds of

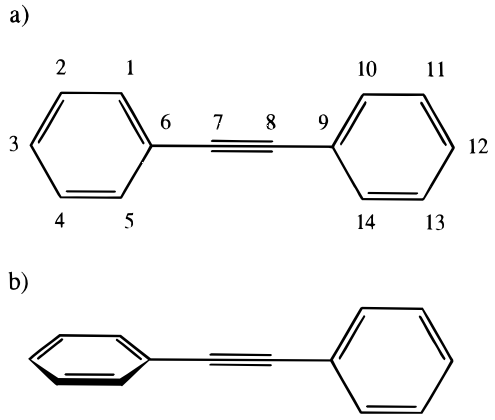

Figure 7. Two forms of 1,2-diphenylacetylene.

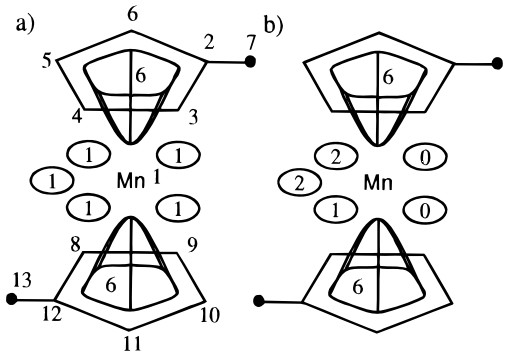

Figure 8. High-spin (a) and low-spin (b) form of 1,1′-dimethylmanganocene. Each manganese atom is bonded to the methylcyclopentadiene anions by coordinative bonds and possesses five π-electron systems that contain the valence electrons of the metal.

transition metals are also strengthened through backdonation of electron density from d-orbitals into higher orbitals of the ligand. It is important to account for all valence electrons to represent coordinative bonds. The metal has s-, p-, and d-electrons, the ligand has s- and p-electrons; these electrons are combined into σ- and π-electron systems. A coordinative bond consists of one or several metal atoms or cations and all atoms and electrons of the ligand that are bonded to the metal center. The valence electrons of the metal atom do not participate in the coordinative bond; they are treated like other free electrons and coded as π-electron systems with up to two electrons per system. Thus, it is possible to distinguish between high-spin and low-spin complexes of a transition metal by distributing the electrons of the s- and d-orbitals differently to the respective π-electron systems.

1,1′-Dimethylmanganocene of Figure 8 illustrates the electron systems used in coordination compounds. The energy difference between its high-spin and low-spin complex is so small that both forms are present in dilute hydrocarbon solutions.[16] The high-spin complex is shown in Figure 8a. Each coordinative bond with six electrons connects one cyclopentadienyl ring to the central manganese atom. All carbon atoms of the ring participate in this bond. The remaining five valence electrons of mangnese are distributed into five π-electron systems. Figure 8b illustrates the low-spin complex. The coordinative bonds are the same as before, but the electron distribution for the s- and d-electrons differs from the one in the high-spin compound. In the low-spin compound two π-electron systems are filled with two electrons: one with one electron and two remain empty.

Mn: 1; C: 2−13

These are the π-electron systems and coordinative bonds of the high-spin form (Figure 8a):

(1 2 3 4 5 6 c; 6) (1 8 9 10 11 12 c; 6) (1 p; 1) (1 p; 1) (1 p; 1) (1 p; 1) (1 p; 1)

These are the π-electron systems and coordinative bonds of the low-spin form (Figure 8b):

(1 2 3 4 5 6 c; 6) (1 8 9 10 11 12 c; 6) (1 p; 2) (1 p; 2) (1 p; 1) (1 p; 0) (1 p; 0)

2.4. Interaction Groups. The concepts presented above only allow the description of covalent bonds. The new model can be extended to describe other interactions such as electrostatic interactions. In order to accomplish this, the concept of interaction groups was introduced. An interaction group combines a set of atoms and/or electron systems that are associated through interactions such as hydrogen bonds or ionic bonds. In the description of a hydrogen bond the interaction group contains the hydrogen atom, the atom it is bonded to and the σ-electron system between them as well as the atom that contributes the free electron pair.

Figure 9 illustrates how the hydrogen bond in the enol form of a 1,3-diketone is represented by a valence bond structure (Figure 9a) and by the new representation (Figure 9b). The σ-electron systems of Figure 9b are indicated by lines; the π-electron system is formed by the two oxygen atoms and three carbon atoms and contains six electrons. All atoms and electron systems that form the hydrogen bond are contained in the gray oval shape. The following atoms and electron systems belong to the interaction group:

[O: 1, 5 H: 6 (5 p; 2) (1 6 s r; 2)]

2.5. Topological Groups. Topological groups combine atoms or electron systems that have common topological features. They may for instance be part of the same substructure or be topologically equivalent. Such information is often needed for the computation of physicochemical properties and for the definition of reaction rules in the reaction prediction system EROS.

An often needed information is the ring information. It is treated as group information in the following way. After the perception of the ring systems all atoms that belong to the same ring are combined into a group of rings. Atoms that are part of adjacent rings may occur in several such groups. Thus, one can easily store that an atom is part of a ring and to which ring it belongs.

2.6. Combination of Molecules. Molecules are composed of atoms that are bonded to each other by covalent bonds, which are represented by three types of electron systems; σ, π, and coordinative. Other interactions can also occur between atoms and electron systems apart from covalent bonds. If these interactions are intermolecular, i.e., between atoms and electron systems of different molecules, the molecules involved are combined into *aggregates*. An aggregate is composed of several molecules and of interaction groups that contain atoms and electron systems from different molecules. These interaction groups indicate interactions between the molecules and are responsible for the formation of an aggregate. Salts, molecules with

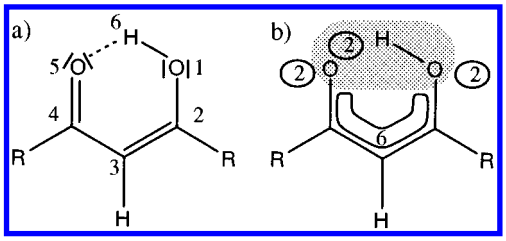

Figure 9. Representation of the hydrogen bond of the enol form of 1,3-diketones: (a) valence bond representation and (b) representation by an interaction group.

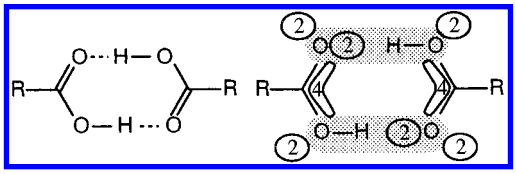

Figure 10. Representation of the dimer of a carboxylic acid as an aggregate. Both interaction groups that are responsible for the formation of aggregates are marked gray.

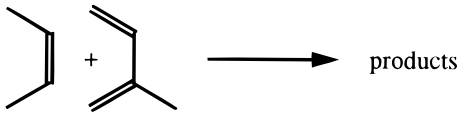

Figure 11. Combination of isolated molecules into an ensemble.

intermolecular hydrogen bonds or catenanes with intertwining ring systems, are candidates for aggregates. Figure 10 shows the formation of an aggregate for the dimer of a carboxylic acid. It contains two carboxylic acids and two interaction groups that indicate the intermolecular hydrogen bonds. The atoms and electron systems of the hydrogen bonds are marked in gray.

The second possibility to combine molecules or aggregates is the *ensemble*. It is a collection of isolated aggregates and/or molecules. Ensembles have no special chemical significance but are useful when several molecules are treated in the same context. Thus, they may combine, e.g., a set of starting materials or products of a reaction as shown in Figure 11. Both molecules needed for a Diels−Alder reaction form the ensemble of starting materials.

For a unique and unambiguous representation of the various species introduced here a hash coding algorith has been developed as an extension of the methods introduced in ref 17. This also allows the perception of constitutional symmetries such as equivalent atoms and electron systems.

3. OBJECT ORIENTED MODEL OF CHEMICAL STRUCTURE REPRESENTATION

The model for the representation of structures presented here was implemented in an object oriented approach. This decision was strongly influenced by the experiences gained during the development of the program systems EROS 6.0, MASSIMO, and FRANZ. These systems had been programmed using procedural languages, such as Fortran and C, with structured programming techniques. The continuing growth of these systems made the management increasingly difficult. Each change or addition to the basic data structures that was introduced to fulfill new needs became increasingly difficult because it demanded a growing amount of changes to the systems at many different locations. The new structure representation developed here could have been incorporated into these systems only by losing efficiency and maintainability.

Therefore, a redesign of the system was performed using an object oriented design to overcome the limitations in extensibility and maintainability of the old systems. The following primary goals led to the decision to build an object oriented class library that can serve as the basic structure representation for the further development of programs assisting in the solution of diverse chemical problems. The resulting library should be easy to use, to extend, and to maintain. This was achieved by exploiting the possibilities of object oriented techniques, such as data abstraction, encapsulation, inheritance, and polymorphism. A brief summary of these concepts is now given.

3.1. Object Oriented Programming. Important concepts of the problem to be solved are searched for during the first stage of object oriented design. These concepts are classified according to common responsibilities or properties.

This process, called *abstraction*, results in the definition of a set of classes that model the composition and behavior of the identified objects. A *class* is composed of two parts: One part is the *public interface* which defines the functionality of the class. It contains all attributes and functions that are available for using the class. The second part is private and hidden from the user of the class. It consists of the internal class structure and the implementation of the defined functions. Thus, the usage of a class is facilitated because the user needs not to learn the details of the implementation but can rely on the fact that each function of the public interface will behave in the promised way. The second advantage of hiding the implementation is the possibility to change it without having to change code that uses the corresponding class. This separation of implementation from the interface is called *encapsulation*.

The design of a class library involves the development of general classes that correspond to general concepts of the problem domain. Such classes are called *base classes*. Specialized *subclasses* are then derived from a base class by using *inheritance*. A base class contains the common aspects, while the derived subclasses extend these with additional attributes or behavior. Thus, inheritance allows the reuse of existing code. The functionality of a base class cannot only be extended, but the implementation of its functions can also be changed for a subclass. This is necessary when special attributes of a subclass have to be taken into account in order to achieve the expected behavior. Such functions are called *virtual functions*. Using virtual functions allows one to prepare existing classes for further extensions without having to modify older code later on. The mechanism to use such functions is called *polymorphism*.

The reader is referred to the relevant literature as given in refs 18−20 for more details on methodology and terminology of object oriented programming.

The library was implemented using the programming language C++. It was chosen in favor of other object oriented languages, because it is rapidly becoming the *de facto* standard for commercial programming. In addition, it allows language mixing with Fortran and C. Thus, useful functions of existing programs can be integrated without having to rewrite them. And finally, low-level programming is also possible when using C++, which allows to tune the performance of time expensive parts of the library, if necessary.

3.2. Hierarchy of Chemical Classes. In a previous chapter the concept of an ensemble was introduced as a

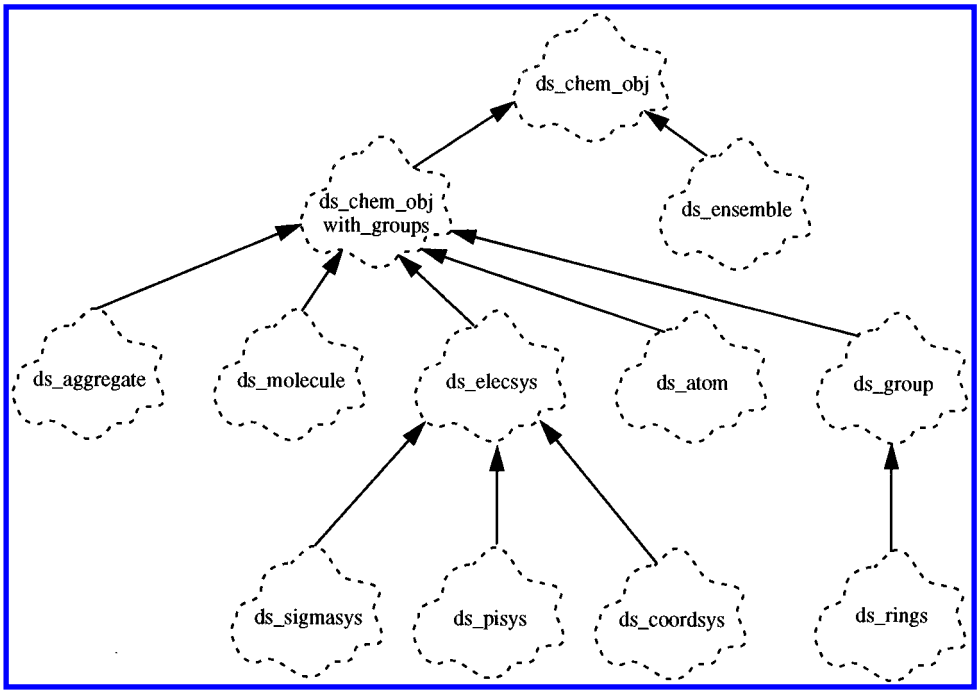

Figure 12. Inheritance hierarchy of chemical classes. The base class is placed at the top of the tree. Each derived class is attached to its base class with an arrow.

chemical entity that is obtained by combining aggregates and molecules. Molecules are composed of atoms and electron systems. These key ideas were now used as a basis for the design of the desired chemical classes. A chemical class was defined for each of the following keywords: ensemble, aggregate, molecule, group, atom, and electron system. Interaction groups and topological groups are both a collection of atoms or electron systems that are related to each other. Therefore, a common class for both types of groups was developed. Objects of all these classes are used as building blocks for the composition of a molecule or an ensemble. The relationships between them form the chemical constitution.

Apart from the constitution of a molecule other properties are often desired. They are derived from the constitution or specified otherwise. Physicochemical and topological properties are such kind of information. They can be atom or electron system attributes or be a property of the entire molecule or ensemble. Therefore, a mechanism for handling such properties that is the same for all chemical classes was developed. This was achieved by using the concept of inheritance. A base class, *ds_chem_obj*, whose public interface consists mainly of functions for property handling is defined. Classes that are derived from it inherit its public interface. For most of the chemical classes also a mechanism for attaching groups must be provided. All classes that can handle groups must have the same interface, which was again defined in a base class. This second base class, *ds_chem_obj_w_groups*, was derived from *ds_chem_obj* publicly and is responsible for the treatment of groups. It possesses a joint interface for property and group handling. Both classes build the foundation for the chemical classes that are used in the applications. All other chemical classes are derived from one of these two base classes. The inheritance hierarchy is shown in Figure 12. The notation of Booch[20] was used to indicate classes and their relationships to each other. All class names have the prefix *ds_* (*d*ata structure) in order to avoid name conflicts in the global namespace that can occur when other libraries are also used.

The base class *ds_chem_obj* is placed at the top of the class tree. Each derived class is shown beneath its direct base class and is connected to it with an arrow. The class *ds_chem_obj_w_groups* is derived from *ds_chem_obj* as explained above. The class *ds_ensemble* is also derived from *ds_chem_obj*, because an ensemble is a collection of unrelated molecules and may not contain any kind of groups. The other classes *ds_aggregate*, *ds_molecule*, *ds_elecsys*, *ds_group*, and *ds_atom*, are all derived from *ds_chem_obj_w_groups*. Objects of classes *ds_aggregate* and *ds_molecule* may contain interaction groups for inter- and intramolecular interactions. Topological groups may be attached to *ds_molecule* objects, if they specify large substructures of the entire molecule, and to *ds_atom* or *ds_elecsys* objects, if the group describes their local chemical environment.

For electron systems and groups some specializations were necessary; the classes *ds_elecsys* and *ds_group* also serve as base classes. One class for each type of electron system is derived from class *ds_elecsys*. One example for a topological group class that is derived from class *ds_group*, *ds_rings* is given in Figure 12.

3.3. Description of Chemical Classes and Their Relationships. Figure 13 shows the basic composition of an ensemble. For clarity, only important relationships between classes are shown. A line is drawn for each class to all other classes to which it is related. The class either references objects of the other class or is referenced by the other class. The filled dots mark the referencing class; the empty squares mark the class that is referenced. The numbers next to the filled dots show how many objects may be referenced. These relationships are explained in the following in more detail together with a description of each chemical class.

A molecule is composed of atoms and electron systems. Each object of class *ds_atom* contains a reference to the molecule and a list of all electron systems it participates in, regardless of their type. Only valence electrons are coded

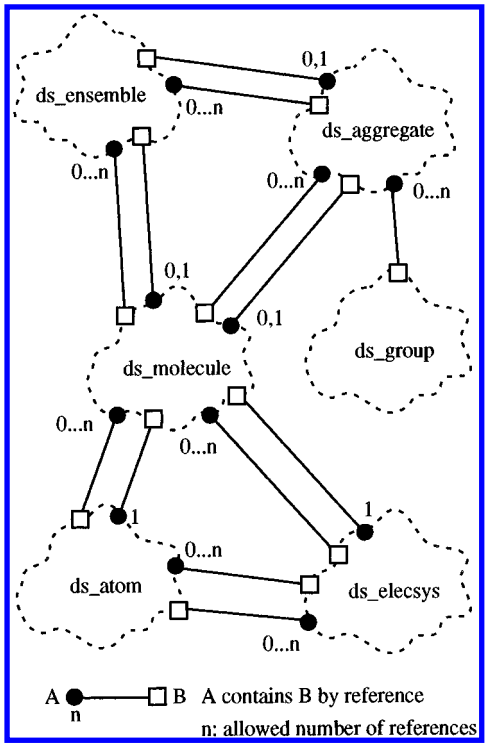

Figure 13. Composition of an ensemble. Classes that reference each other are linked with lines. A black dot denotes the referencing class; an empty square denotes the class that is referenced. The numbers next to each black dot show the number of objects that may be referenced.

using electron systems. All core electrons are regarded as not reactive and are therefore an implicit part of an atom. The attributes that are necessary to define an atom are the chemical symbol, an index, and a label for identification. The public interface allows handling of electron systems.

The representation introduced here divides electron systems into different types: σ-electron systems, π-electron systems, and coordinative bonds. All have common attributes that are specified in a class *ds_elecsys*. Individual classes for all types of electron systems are then derived from *ds_elecsys*. An object of class *ds_elecsys* contains a reference to the molecule it belongs to and a list of all atoms that are part of it. Its attributes are the number of electrons localized in the system, an identifier for the system type and an index for identification. The public interface includes functions that deal with changes of atoms and the number of electrons in a system. The class for σ-electron systems, *ds_sigmasys*, has in addition to *ds_elecsys* information about the geometry of the system, whether it is a regular two-electron, two-center σ-electron system, or a closed or bridging multicenter σ-electron system. If there is a central atom in a bridging σ-electron system, it is also specified. Neither the class *ds_pisys* that handles the treatment of π-electron systems nor the class *ds_coordsys* that stores coordinative bonds have additional attributes. However, they are needed to model different behavior of these electron system types in the course of a reaction. The class *ds_molecule* contains lists with all atoms and electron systems. The public interface offers functions to add and remove atoms and electron systems. The connectivity information is implicitly given by the types of electron systems used to build a molecule. The class *ds_molecule* also offers methods to access all neighboring atom pairs. The stereochemistry information of atoms and electron systems is not part of their classes because it is not essential to define a molecule. It will be added in the future as optional properties of atoms and electron systems.

Objects of class *ds_atom*, *ds_elecsys*, and *ds_molecule* reference each other more often (Figure 13) than is necessary to completely represent a molecule. These additional references allow a more efficient access to atoms and electron systems. If each object of class *ds_atom* or *ds_elecsys* were only referenced once in a molecule, specific access to all electron systems of an atom and vice versa would always need a full search of the entire list of atoms or electron systems. The same applies to the back references of atoms and electron systems to their molecule.

The class *ds_aggregate* is derived from *ds_chem_obj_w-_groups*. An object of this class consists of a list of molecules and a list of groups that contain the interactions between the molecules in this aggregate.

Molecule and aggregate objects may be isolated entities or parts of ensembles. If they are part of an ensemble, they contain a reference to the ensemble object they belong to. The class *ds_ensemble* consists of a list of aggregates and/or molecules. Its interface allows access to all aggregates and molecules.

The class *ds_group* consists of lists of atoms and electron systems and has a public interface that allows their handling. If an interaction group is to be implemented, a property that specifies the strength of the interaction also has to be defined for such a group. If one wants to define topological groups that collect atoms or electron systems with special topological features, it is possible to derive a specialized group class from *ds_group*. The specialized group class may then contain a function to find all atoms or electron systems necessary to build the group. These new classes can be integrated into existing applications without additional work because the functions for finding all atoms in a specialized group are virtual. They are dynamically linked to the application and need not be known at the time of the design of the library. In addition, the mechanism for handling groups is independent of the type of groups. We will show here that the design of a class library using object oriented programming is especially suitable for the implementation of the structure representation by the following example. It is sometimes necessary to be able to represent a molecule also in a valence bond representation in order to be compatible to other structure representation systems. In order to accomplish this task, the potentials of object oriented programming can fully be exploited. Thus, it is not necessary to design new classes for the representation of molecules and ensembles. Most existing classes can easily be reused because the composition of a molecule does not depend on the types of electron systems or atoms that are used to build a molecule. It is therefore possible to derive specialized classes for valence bond atoms and bonds and to use objects of these classes when constructing the desired molecules. The specialized class for atoms, *ds_vbatom*, may then contain additional attributes and functions to handle free electron pairs; the specialized class for electron systems, *ds_vbbond*, may possess additional functions that allow the treatment of bond orders. Figure 14 shows the relationships between the classes that define a molecule in a valence bond representation. All other classes presented in Figure 13 are not affected by these changes.

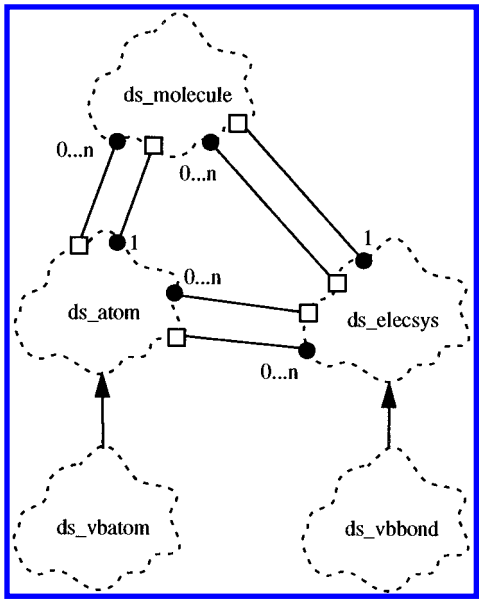

Figure 14. Reuse of the class library for the representation of molecules in a valence bond model.

3.4. Treatment of Properties and Groups. In the EROS 6.0 reaction prediction system a reaction is only carried out if the atoms and electron systems of the reaction center fulfill certain conditions. Such conditions may be topological or physicochemical properties of the involved atoms, bonds, or molecules. They are derived from the constitution of the molecule and must have values in a given range. The systems for prediction of infrared spectra[21] and those for QSAR studies also use these properties as descriptors for molecules. A set of properties is defined and implemented in the EROS 6.0 system. However, it is difficult to extend this set without making major modifications to the entire system. The interface for property handling in the new library was designed having two goals in mind. One goal was the possibility to extend the set of accessible properties without having to change the core system and ensuring that the access methods are also valid for newly defined properties. The second goal was to combine the retrieval of these properties with dynamic recalculation after changes to the chemical objects have been made. Additionally, a new type of property is added which is not calculated from the chemical constitution. It is used to attach tags to chemical objects that allow their tracing during a reaction sequence. For instance tracing of isotope labels of atoms can be treated this way.

The property handling mechanism is composed of three parts: definition, status information, and access to the values. Attributes have to be defined for each property in order that it can be used. A class *ds_prop_definit* has been introduced to store these attributes. It is composed of an identification string, an unique entry number for access, a function that calculates the property values, a default value, and conditions when the property will become invalid. These conditions depend on the type of modification to be performed on an ensemble. They range from invalidation through small changes like merging two separate molecules into one ensemble to no invalidation at all, even after changes have been made to atoms and electron system during a reaction.

Status information is stored in objects of class *ds_pg_status*. It consists of a boolean value for its validity. The property values of atoms or electron systems are calculated for one entire molecule in one step. Hence, the status information is held only once for all atoms or electron systems in a molecule. Each molecule, aggregate, group, and ensemble is responsible for its own status information. The values for all properties are stored with each chemical object. They are accessed and changed by functions of class *ds_chem_obj*. These functions are also responsible for the automatic recalculation of property values after changes have been made to an ensemble that result in an invalid status. An increase in the number of available properties can be achieved in two ways. Either the array with all known property definitions is extended and linked to the program, or new property definitions may be created during a program run and dynamically added to an extendable container that holds all property definitions available for a specific application.

Groups are treated in a similar manner. Their handling is also separated into definition, status information, and access. It differs only in how groups of atoms or electron systems are treated. They are not collectively derived for all atoms or electron systems of one molecule but separately for each of them which is more flexible. Their access is managed via functions of class *ds_chem_obj_w_groups*. They also control the automatic derivation after changes that result in invalid groups.

4. CONCLUSIONS

The presented model of structure representation allows the handling of a wide range of chemical species. Inorganic and organometallic compounds as well as nonclassical cations or other reaction intermediates with multicenter bonds can be represented and are thus amenable to information handling systems such as database handling, substructure searches, etc. Furthermore, the ambiguities in the representation of organic compounds that contain delocalized or orthogonal π-systems are resolved. This structure representation replaces a connection table representation that only allows one to describe chemical structures containing bonds that can be localized between pairs of atoms. The model separates the σ- and π-part of multiple bonds into two bond types and introduces a third bond type for the formation of coordinative bonds. The restriction of having to localize bonds between pairs of atoms has been abolished. Electrons may be delocalized between more than two atoms in all three bond types.

Object oriented programming was used to implement these concepts. The chemical concepts that were introduced here were translated into classes that model the constitution and behavior of molecules. A flexible class library is provided that serves as a basis for diverse chemical applications. During the design of chemical applications it is now possible to use chemical language which simplifies the process of finding computational solutions for chemical problems.

It was shown that the application of object oriented programming also greatly enhances the maintainability and simplicity of usage of the developed library. In addition, the library is open to be adapted to other models of structure representations that are currently not supported.

A first application example of this structure representation is its use in the redesign of the reaction prediction system EROS. The new features are an integral part of the core system to generate reactions and make it amenable to model

reactions that involve inorganic or organometallic compounds.[22]

Future work will concentrate on the extension of the representation with descriptors for stereochemistry and the extension of the set of available physicochemical properties.

ACKNOWLEDGMENT

We wish to thank R. Höllering and Dr. K.-P. Schulz for many fruitful discussions. This work was partly supported by the Bundesminister für Bildung, Wissenschaft, Forschung und Technologie of the Federal Republic of Germany.

REFERENCES AND NOTES

(1) *Chemical structure systems. Computational techniques for representation, searching, and processing of structural information*; Ash, J. E., Warr, W. A., Willett, P., Eds.; Ellis Horwood: Chichester, 1991.

(2) Dalby, A.; Nourse, J. G.; Hounshell, W. D.; Gushurst, A. K. I.; Grier, D. L.; Leland, B. A.; Laufer, J. Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 244−255.

(3) Ugi, I.; Gillespie, P. Beschreibung chemischer Systeme und ihrer Umwandlungen durch *be*-Matrizen und ihre Transformations-Eigenschaften. *Angew. Chem.* **1973**, *83*, 980−981. Representation of Chemical Systems and Interconversion by be Matrices and their Transformation Properties. *Angew. Chem., Int. Ed. Engl.* **1973**, *10*, 914−915. (b) Ugi, I.; Gillespie, P. Stoffbilanz-erhaltende Sysntheseswege und semi-empirische Syntheseplanung mittels elektronischer Datenverarbeitung. *Angew. Chem.* **1973**, *83*, 982−985. Matter Preserving Synthetic Pathways and Semi-Empirical Computer Assisted Planning of Syntheses. *Angew. Chem., Int. Ed. Engl.* **1973**, *10*, 915−919.

(4) Gasteiger, J. A Representation of Pi-Systems for Efficient Computer Manipulation. *J. Chem. Inf. Comput. Sci.* **1979**, *19*, 111−115.

(5) Gasteiger, J. Automatic generation of pericyclic reactions. *Z. Naturforsch.* **1979**, *34b*, 67−75.

(6) Mockus, J.; Stobaugh, R. E. The Chemical Abstracts Service Chemical Registry System. VII. Tautomerism and Alternating Bonds. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 18−22.

(7) Welford, S. M. The Beilstein Data Structure for Organic Compounds. In *Software-Entwicklung in der Chemie 1*; Gasteiger, J., Ed.; Springer-Verlag: Berlin, 1987; pp 5−11.

(8) Dugundji, J.; Ugi, I. An Algebraic Model of Constitutional Chemistry as a Basis for Chemical Computer Programs. *Top. Curr. Chem.* **1973**, *39*, 19−64.

(9) Ugi, I.; Stein, N.; Knauer, M.; Gruber, B.; Bley, K.; Weidinger, R. New Elements in the Representation of the Logical Structure of Chemistry by Qualitative Mathematical Models and Corresponding Data Structures. *Top. Curr. Chem.* **1993**, *166*, 199−233.

(10) Stein, N. New Perspectives in Computer-Assisted Formal Synthesis Design. Treatment of Delocalized Electrons. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 305−309.

(11) Dietz, A. Yet Another Representation of Molecular Structure. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 787−802.

(12) Röse, P.; Gasteiger, J. Automated Derivation of Reaction Rules for the EROS 6.0 System for Reaction Prediction. *Anal. Chim. Acta* **1990**, *235*, 163−168.

(13) Gasteiger, J.; Ihlenfeldt, W. D.; Röse, P. A Collection of Computer Methods for the Synthesis Design and Reaction Prediction. *Recl. Trav. Chim. Pays-Bas* **1992**, *111*, 270−290.

(14) Gasteiger, J.; Hanebeck, W.; Schulz, K. P. Prediction of Mass Spectra from Structural Information. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 264−271.

(15) Schulz, K. P.; Bauerschmidt, S.; Höllering, R.; Gasteiger, J. Automatic Elucidation of Reactions in the Mass Spectrometer. In *AIP Conference Proceedings 330*; Bernardi, F., Rivail, J. L., Eds.; American Institute of Physics: Woodbury, NY, 1995; pp 726−733.

(16) Switzer, M. E.; Wang, R.; Rettig, M. F.; Maki, A. H. On the Electronic Ground States of Manganocene and 1,1′-Dimethylmanganocene. *J. Am. Chem. Soc.* **1974**, *96*, 7669−7674.

(17) Ihlenfeldt, W. D.; Gasteiger, J. Hash Codes for the Identification and Classification of Molecular Structure Elements. *J. Comput. Chem.* **1994**, *15*, 793−813.

(18) Martin, R. C. *Designing Object-Oriented C++ Applications Using the Booch Method*; Prentice-Hall: Englewood Cliffs, NJ, 1995.

(19) Cline, M. P.; Lomow, G. A. *C++ FAQs*; Addison-Wesley: Reading, MA, 1995.

(20) Booch, G. *Object-Oriented Analysis and Design with Applications*; Benjamin/Cummings: Redwood-City, CA, 1994.

(21) Schuur, J. H.; Selzer, P.; Gasteiger, J. The Coding of the Three-Dimensional Structure of Molecules by Molecular Transforms and Its Application to Structure−Spectra Calculations and Studies of Biological Activity. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 334−344.

(22) Höllering, R. Unpublished results.

CI9704423