As part of the StAR project, for the design of a computer system to support risk assessment, it has been necessary to develop a graphical language for the representation of generic structures. These structures are used by rule writers to describe toxicophores for risk assessment. Until now, the input of generic structures in knowledge-based systems has very often been by means of a SMILES-like linear notation. The new StAR graphical language allows the use of a wide range of Markush features including atom lists, bond lists, G-groups, and superatoms. In addition, the StAR graphical language allows rule writers to input rules easily to the knowledge base via a user-friendly graphical interface. This language is well-suited for the representation of chemical features required for expert general toxicity systems, chemical reaction systems, and substructure database systems.

INTRODUCTION

Expert systems, computerized systems which mimic the thinking and reasoning of human experts, are increasingly being used in biochemical and chemical applications, e.g. to assess the toxicity of compounds. These systems use sets of rules to make predictions in novel situations rather than relying on complex calculations. Another area of future promise is their use to predict the metabolic transformations a compound is likely to undergo in living entities. All these knowledge-based systems used in chemistry have in common that they apply structure–activity relationships to chemical compounds. Knowledge-based systems all require the input, manipulation, and pattern matching of chemical graphs or substructures. In the case of Computer Assisted Organic Synthesis (CAOS) programs they link the presence of reaction patterns (or retro-reaction patterns called retrons) with specific reactivities; in the case of toxicity prediction system they associate a chemical substructure with a particular toxic endpoint.

RULE DERIVATION AND CHEMICAL PATTERN REPRESENTATIONS

Both human experts and computer systems are capable of deriving from specific information (or facts) the rules to be included in the knowledge base. This mainly involves the derivation, from a set of congeneric structures, of the common substructure responsible for inducing the studied activity. The rule induction process is based on the reasonable supposition that compounds have similar biological activities or chemical reactivities, because they possess common structural features (Figure 1). Most computer rule induction systems identify toxic/active substructures by systematic generation of common substructures or statistical analysis of fragment distribution in active and inactive compounds. Once these structural features have been identified, the human expert can generalize these in the form of a generic structure, also called a Markush structure or pattern. Markush structures, containing generic parts, represent a set of specific structures. The generic structure can be of variable complexity, ranging from a simple substructure to a highly parametrized structure.

In most knowledge base systems, chemical patterns are represented, as far as the rule writer is concerned, by a character string following SMILES-like notation or even in some cases by combinations of built-in generic terms. DEREK, a system for toxicity prediction, and LHASA, a system for synthesis planning, both used a SMILES-like notation called PATRAN for the coding of chemical patterns in their knowledge base (Figure 2). To complete PATRAN patterns, an English-like language called CHMTRN allows the association of structural inquiries to refine the structure specificity of rules (Figure 3).
The coding of the chemical structures is effected by a human expert before being input in the system knowledge base. It is understandable that rule writers are looking for more friendly and appropriate tools for building and maintenance of knowledge bases. Another limitation of these traditional languages comes from the fact that they are designed to exclusively represent organic compounds. The manipulation of chemical knowledge, especially in the risk assessment context, is not only limited to organic chemistry but can also be applied to inorganic chemistry.

THE STAR PROJECT

The end of the century is marked by increasing regulations, controls, and developments of standards. Consequently, risk assessment, e.g., evaluation of chemical toxicity, has become an important factor in policy and safety decision making. Expert systems for toxicity prediction such as DEREK are valuable tools to organizations and individuals for evaluating hazards associated with particular situations or products.

Computers are particularly good at processing large amounts of data and at reaching logical conclusions from a set of arguments. In the risk assessment context, data are made up of sets of experiences or experiments. These data, e.g., data on confirmed carcinogens, often need reworking by a human expert before being input in the system knowledge base.

An argument is an assessment derived by the system from its sets of rules in a given context. The communication of details of its reasoning and of its assessment can be a difficult matter especially when addressing nonexperts of the domain.

The StAR project intends to address three major problems faced by computer systems for toxicity prediction:

1. identifying and considering the arguments for and against a proposed risk to reach an overall impartial assessment,
2. communicating its conclusions about risk and reasoning in an unambiguous form understandable by any individual,
3. facilitating knowledge input by providing the rule writer with tools which make the writing of rules quick and easy.

In order to reach and communicate an assessment of risks, StAR utilizes the Logic of Argumentation which has been described elsewhere by Krause et al. StAR uses general rules from the domain of toxicity combined with rules relating toxicity to the presence of specific structural alerts. In the chemical context, the knowledge base of the risk assessment system is constituted of structure-activity rules aimed at predicting the toxicity of chemicals. To facilitate the knowledge representation, StAR makes extensive use of the structural features of chemical representations. It provides user-friendly interfaces to render the input, editing, and correction of this structural information, simple, fast, and natural. As described in this paper, StAR provides the rule writer with three essential tools to facilitate the knowledge manipulation: a graphical language to help the expert formulating the structural data in a way suitable for rule formulation and knowledge base input; a chemical editor for drawing in structural patterns that are components of structure-activity rules; a rule editor for the production of the structure-activity rules. Adaptable to domains other than
CHEMICAL STRUCTURE PROCESSING

Standard perceptions, e.g., cycle and aromaticity perceptions, performed by chemical knowledge base systems on chemical structures have been the object of many publications. They extensively make use of the fact that chemical structure diagrams are mathematical graphs. More of a problem are the operations a chemical expert system has to perform on Markush structures contained in its knowledge base. All the features of the structural language must be “understood” by the expert system in order to be utilized in the processing phase, i.e., assessment of chemical toxicity in our context. For example, the use of superatomsm relies on the perception by the system of the corresponding structural features or chemical properties.

Pattern matching is at the heart of the knowledge processing. The expert system routinely matches the generic structures in its knowledge base on specific structures. The pattern matching of Markush structures on chemical compounds is a difficult problem that has been studied for years. The matching problem is a serious limitation to the use of Markush structures in expert systems. If not restricted to reasonable generic structures, the matching of patterns can quickly become complex and even unmanageable. Within the StAR project, we have limited the complexity of the structural language to a reasonable but sufficient level. For example, we do not allow the definition of fragments within fragments.

THE STARS YSTEM

The StAR structural language describes generic organic and inorganic chemical structures by graphical means. One important aspect of the language is that the structures that are graphically-entered generate patterns coded on Molfiles directly understandable by the StAR system, i.e., substructures which StAR can use for pattern matching. The Psi graphical interface used by StAR is an extension of the drawing package in STN express developed by Hampden Data Services [Hampden Data Services Ltd., 49 Stoney Street, Nottingham NG1 1LX, UK].

The language includes five important features to allow the graphical definition of accurate structure—activity patterns: the generic definition of atoms and bonds; the use of the NOT operator to refine atomic environment; the definition of lists of atoms, bonds, and groups; the use of the OR, NOT, and AND operators; and the use of variable points of attachment. The generic definition of atoms and bonds allows the description of families of compounds. If required, patterns can be more precisely described by using the NOT operator to pose conditions on the environment of specific atoms. The definition of atoms, bonds, and groups in the form of lists allows the general description of substructures dissimilar by one functional group, atom, or bond type. In addition, the StAR structural language offers the possibility of combining the item definitions in using the OR, NOT, and AND operators. Finally, to allow the description of Markush structure of position, the language will also support atom or list of atoms or groups, variably attached at more than one point of the pattern. A detailed description of the StAR structural language is given as an annex to this paper.
STAR KNOWLEDGE BASE EDITOR

The StAR knowledge base editor allows the rule writer to describe structural alerts, i.e. substructures considered responsible for activity induction, and to formulate structure—activity rules electronically. The Psi chemical editor accepts any Markush structure defined in the StAR structural language. The rule writer can use this chemical editor to put pattern structures into the system and also to check and to edit patterns already in the database (Figure 4). Once the pattern is stored, the rule writer links it with a specific activity by entering a rule, e.g., “IF pattern #x THEN activity #y”. The rule editor, specifically customized for the need of the StAR risk advisor, presents a series of menus making the creation of such rules fast and easy (Figure 5).

STAR PROJECT RESULTS

Started three years ago, the StAR project is now near completion. A carcinogenicity risk advisor has been implemented on a PC platform using the StAR technology. The StAR structural language has been used to represent the chemical structures (or structural alerts) stored in its database. A chemical editor and a rule editor have also been implemented. By using the chemical editor we have been able to graphically enter a series of active patterns. We have used the rule editor to link these structural alerts with toxic endpoints. The first version of the StAR carcinogenicity risk advisor is now linked with a limited but accurate knowledge base. It can already be used to assess the carcinogenicity of certain classes of compounds. For example, Figure 6 shows the alert summary and the StAR report obtained when processing a query structure. The system found evidence both for and against the realization of human carcinogenicity. The system has detected the presence of a structural alert for peroxisome proliferator activity in the query structure, but other rules in the knowledge base questioned the relevance of this activity to Humans.

CONCLUSION

The features of the StAR structural language have proved to be appropriate for the representation of structural alerts stored in the prototype version of the carcinogenicity risk advisor. In this prototype, the combination of the structural language, the chemical editor and the rule editor have eased the chemical knowledge manipulations as expected. Work is now under way to apply these graphical tools to the constitution of a full-scale knowledge base for carcinogenicity risk assessment.

The structural language developed within the scope of the StAR project is particularly appropriate for the representation of structural alerts. Combined with graphical computer tools, it provides a promising solution for chemical knowledge manipulation. As such, this model could be used at the core of other types of systems that require chemical knowledge manipulation and representation. With a few additions, the technology could be used for the implementation of databases of Markush substructures searchable by specific compounds or of databases of structures searchable by Markush substructures. Another possible application is the adaptation of the StAR structural language to represent the transformation patterns used in systems for Computer Assisted Designed Organic Synthesis (CADOS). Reaction editors could then be used to manipulate reaction information graphically. Such
tools would be of great help to control the huge amount of knowledge processed by these systems. Reaction systems are by no means limited to CADOS; reaction prediction systems, e.g., metabolism prediction systems, could also benefit from this approach.21

ACKNOWLEDGMENT

The StAR project is supported by the Department of Trade and Industry and the Engineering and Physical Sciences Research Council.

APPENDIX: THE STAR STRUCTURAL LANGUAGE

1. Atoms and Bonds. Items fundamental to the structural language include simple or generic definitions of atoms and bonds. Atoms and bonds can be generically defined as a set of specific items, for example, halogen atom representing chlorine, bromine, fluorine, and iodine or any type of bond including the set of all specific bonds.

1.1. Atoms. 1.1.1. Specific Atoms. A specific atom is by definition one of the basic atom type of the periodical table. Note that hydrogens can be explicitly specified. As the patterns that are entered are to be considered as substructures, explicit hydrogens are meaningful and therefore kept.

1.1.2. Generic Atoms. Generic atoms represent sets of atoms or atomic groups with similar properties. These properties can be chemical, e.g., aromaticity or structural, e.g., alkyl group. The language not only allows commonly-used generic atoms, metals, halogens, aromatic atoms, heteroatoms, and alkyls, but also allows less common generic entities of great utility for pattern description, non-hydrogen atoms, any atom types, nonfunctionalized atoms, etc. This list is expandable (and expanding) on request by the rule writers (Figure 7).

List of generic atoms currently available:
“X”: halogen atom
“A”: any atom except H
“Q”: any atom except C, H (heteroatoms)
“M”: metal
“Ar”: aromatic atom
Alkyl1, Alkyl2, Alkyl3, Alkyl4: “alkyl” groups

It is also possible to give attributes to atoms such as “in chain”, “in ring”, “at ring junction”.

About Alkyl Groups. The elusive nature of what rule writers define as “alkyl” deserves some explanation. The chemist is generally quite clear about what “alkyl” means, a saturated hydrocarbon group. However, when asked what it should mean, rule writers say that it depends on the case. On this account, the StAR language currently offers four “alkyl” group definitions:

a. “Alkyl1” is the alkyl group in the strict sense of the term: a saturated hydrocarbon group.
b. “Alkyl2” is a saturated or unsaturated hydrocarbon group.
c. “Alkyl3” is a group with no functionalities on the atoms at and α to the anchor point.

Figure 6. The StAR Carcinogenicity Risk Advisor can already be used to assess the carcinogenicity of a limited class of compounds.

...
The use of generic atoms in patterns allows the description of families of compounds.

Example of a pattern containing a generic bond defined as aromatic.

Example of a pattern containing a G-group defined by a list of two fragments.

Patterns including restrictions on the environment of specific atoms.

d. “Alkyl$_1$” is a group with no functionalities, other than multiple C–C bonds, on the atoms at and α to the anchor point.

1.2. Bonds. 1.2.1. Specific Bonds. The list of bonds currently available includes those commonly used, single, double, and triple bonds, but the list is to be extended to those bonds more specific to inorganic chemistry: dative, ionic, and π bonds.

1.2.2. Generic Bonds. Apart from specific bond types, a number of generic bond types are also defined grouping bonds with similar chemical or structural properties: aromatic, any type, and ring junction (Figure 8). As for the atom type set, the set of generic bond types available will be expanded when needed. It is also possible to give attributes to bonds such as “in chain” or “in ring”.

1.3. G-Groups. Typically, a G-group is a fragment (or substructure) associated with two anchor points, one on the fragment attachment point and one on the pattern (G-atom position). The two anchor positions define how the fragment is attached to the main pattern. The possibility to define lists of fragments is a powerful mean for Markush structure representation and is therefore of very common usage (Figure 9) (see section “3. Lists”). The use of generic atoms, e.g., alkyls, aromatic, etc., is allowed in G-group definitions. However, only one level of G-group definition is supported, i.e., it is not permitted to nest a G-group in another G-group definition. This restriction is intended to limit the complexity of the Markush structures to keep the internal structure processing of the system manageable.

2. NOT Operator. Often, the expert applies structural restrictions to general patterns to obtain more accurate pattern descriptions, e.g., by restricting the number of hydrogens on an atom. The StAR structural language allows imposition of conditions on the attributes of specific groups (Figure 10).

3. Lists. The definition of atoms, bonds, and groups in the form of lists allows the general description of substructures dissimilar by (at least) one functional group or bond type (Figure 11).

4. Operators. The StAR language will support the combination of OR and NOT operators to accurately define types and item properties (Figure 12).

5. Variable Points of Attachment. The presence or absence of a specific atom at precise locations of a pattern structure can have dramatic effects on its activity. Positional isomers can present similar activities, e.g., o-, m-, and p-chlorotoluene can be all active. In such situations, to avoid splitting the pattern (e.g., one for each positional isomer), the StAR structural language supports the use of variable points of attachment (Figure 13).

REFERENCES AND NOTES


