VET: A Tool for Reaction Plausibility Checking

Joseph L. Durant,* Burton A. Leland, and James G. Nourse

Elsevier MDL, 2440 Camino Ramon, Suite 300, San Ramon, California 94583

Received September 9, 2005

Production of chemical reaction databases is a multistep process, with the possibility of errors at each of these steps. VET is a tool developed to trap errors in the chemical reactions identified as a part of this process. VET has been designed to minimize the acceptance of incorrect reactions, while still supporting various common practices in reaction depiction, including unbalanced reactions, suppressed components, and reactions with alternative products. We discuss the assumptions made in its construction, a general overview of its structure, and some performance characteristics.

INTRODUCTION

Construction of reaction databases often involves the extraction of chemical reactions from textual descriptions. As part of this process chemical names are identified, extracted from text, and converted to structures. Reaction roles are assigned to these structures, and other ancillary information is identified and extracted. Errors can occur in each of these activities, leading to errors in the resulting database entries.

Trapping such errors is a formidable but important task. We will focus our attentions here in trapping errors in extraction of the reaction components and the associated extracted reactions. Such extracted reactions are often nonstoichiometric, reflecting lack of explicit stoichiometry in the source document. Likewise, the extracted reactions often do not contain all the reaction components. This can reflect omission of "obvious" reaction components in the source document or the fact that a named compound may be present in more than one role (reactant and solvent, for example). The extracted reactions can also represent reactions with alternative products, again reflecting the description present in the source document.

Such "nonideal" descriptions arise naturally and are desired because they emphasize what is chemically interesting and suppress what is judged insignificant or obvious. This enhances the clarity of the chemical ideas being presented but considerably complicates detection of errors.

There is a long history of interest in the automated treatment of chemical reactions. Of some interest in the present context are techniques for the storage and recovery of reactions from databases.

Solutions enabling the storage and searching of reactions in databases are varied, but at their beginnings they all rely on the ability to automap the reaction, i.e., to establish atom−atom correspondences between reactants and products, which also allows one to establish bond roles (made, broken, changed order, unchanged).[1−4] This automapping can be used directly in structure searching[5] to set keys which are then used for similarity searching[6] or to generate a reaction classification.[7]

VET was developed to detect errors in chemical structure and role assignment in the production of reaction databases. VET is coded in Cheshire,[8] Elsevier MDL's chemical parsing and manipulation language. It consists of a number of pragmatically developed filters and checks which address problems common in the automated reaction extraction used in production of the Patent Chemistry Database.[9]

OVERVIEW

Design Goals. A principle design goal of VET is the minimization of the acceptance of incorrect reactions. Accepting an incorrect reaction is considered to be a much more serious mistake than incorrectly rejecting a correct reaction. In this way database quality is maximized; rejected reactions can be reprocessed, either by humans or by improved versions of VET, for eventual inclusion into a database.

A second goal is to support real world usage patterns in reaction representation. As part of this we explicitly treat three common patterns, which are depicted in Figure 1. These patterns are as follows: (a) use of an unbalanced (nonstoichiometric) reaction depiction, (b) suppression of "uninteresting" components, either reactants or byproducts, and (c) use of a multichannel reaction depiction, aggregating alternative reaction products into a single reaction equation.

Finally, VET was designed as much as possible to be modular, with the ability to turn individual filters and checks on and off.

Overall Design. At its core VET is an automapper. If an acceptable automapping can be generated the reaction is accepted. If an acceptable automapping cannot be generated the reaction is rejected.

There are a number of criteria used to assess the acceptability of a given automapping. A scoring system has been implemented to allow for gradations in reaction acceptability.

As a practical matter VET performs a preliminary filtering of proposed reactions. This filtering is mainly directed at detecting a number of failures arising in extraction of chemical components from text. This preliminary filtering improves VET's overall performance by eliminating costly evaluation of patently incorrect reactions.

* Corresponding author e-mail: J.Durant@mdl.com.

10.1021/ci050390e CCC: $33.50
Published on Web 02/18/2006

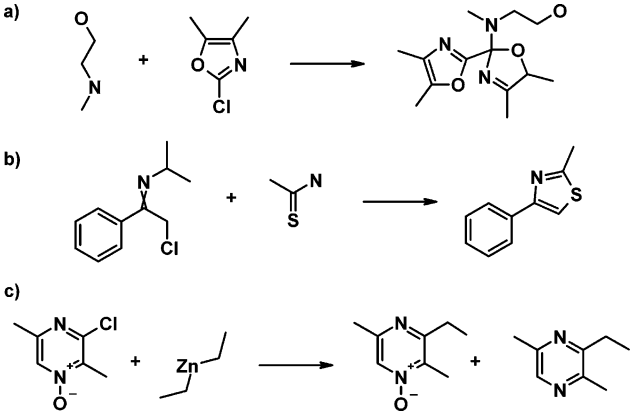

Figure 1. Examples of (a) unbalanced reaction, (b) reaction with suppressed components, and (c) a reaction with alternative products.

Next an automapping of the reaction is attempted. This automapping undergoes a series of basic checks for feasibility and may be redone if the initial automapping is rejected. Various strategies may be employed to improve these subsequent automappings.

Once an automapping has been found which passes the basic tests, VET performs a set of more advanced checks. These checks focus on the presence and type of unmapped atoms in the reaction. These final checks are focused mainly on trapping errors arising from misidentified reaction components or garbled reactions.

Implementation. *Overview of Cheshire.* VET is implemented as a script in Cheshire, Elsevier MDL's proprietary chemical scripting language.[8] Cheshire is designed to support chemical structure analysis, interpretation, and manipulation. Cheshire abstracts chemistry into the Cheshire environment and puts interaction with the compute environment, such as filesystem access and I/O, into the calling application. This abstraction allows Cheshire scripts to be run in any supported hardware/software environment without alteration. As an example of this we consider VET. VET was developed using a Perl-based test and development program on a Linux platform. The Perl code handles reading and writing to the console and reading and writing reaction files to disk. Strings containing the VET script and strings containing molfiles or rxnfiles[10] are sent into the Cheshire environment where the VET script is executed. The Cheshire environment passes strings containing script output, molfiles, and rxnfiles back to the Perl application. Production runs accessed Cheshire on a Solaris platform using sockets interacting with Cheshire's C interface and used the unaltered VET script. A wide variety of interfaces allow Cheshire to be called from a variety of programming environments.

Syntactically the Cheshire language is modeled on JavaScript. Basic Cheshire objects include strings, molecules, reactions, collections, and iterators. There are 29 atom and bond properties, together with 22 structural (Sgroup[11]) properties and a number of other miscellaneous properties. Properties can be retrieved with List(); collections containing desired property values can be collected with a Find() operation. The Set() method can be used to change properties on members of a collection. Iterators allow collection contents to be accessed sequentially, and decision and flow control statements allow complex programs to be constructed.

Additionally Cheshire provides access to other functionality present in various Elsevier MDL products. There are Cheshire methods to perform substructure mapping, flexible structure matching, MDL key generation, and registration checks. In all there are approximately 170 Cheshire methods.

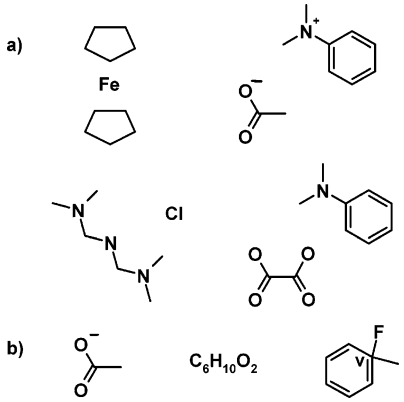

Figure 2. (a) Examples of accepted multifragment reaction components and (b) examples of unacceptable single-fragment reaction components.

In the case of VET, a Cheshire environment is created, the VET script is loaded, and then successive proposed reactions are loaded, together with any additional molecules identified as related to the reaction. The VET script is then executed, and a return string and, if appropriate, a transformed reaction are returned.

Preliminary Filtering. We begin by ensuring that both reactants and products have been identified in the extracted reaction. Reactions with no reactants or no products are rejected.

Sometimes chemical names are incorrectly parsed. This can result in multiple compounds being considered a single component or in only part of a compound being present in a component, for example, the "benzoate" in "methyl benzoate".

To trap the first class of these errors VET filters multifragment components. Acceptable classes of multifragment components include components containing radicals and components containing two uncharged fragments: one a proton donor and one a proton acceptor. We also accept components containing single heavy-atom fragments where the fragment is a proton donor. Finally, VET accepts components containing charged fragments where both + and − charges are present. The charges do not have to be balanced.

Trapping errors of the second class is more straightforward. VET filters single fragment components and rejects those with only + or only − charges.

Idiosyncratic failures in name-to-structure converters have produced erroneous structures with explicit valences and structures lacking bonds. Both of these are also easily detected and subsequently rejected by VET.

Examples of components accepted and rejected by this prefiltering can be seen in Figure 2.

Automapping and Basic Checks. The automapping portion of VET utilizes a custom automapper, the Cheshire automapper,[4] and Cheshire scripts to provide a more robust reaction automapping than would be possible using any of the tools singly. Additional Cheshire scripts are used to assess the accuracy of the automapping and, if necessary, drive additional automapping steps.

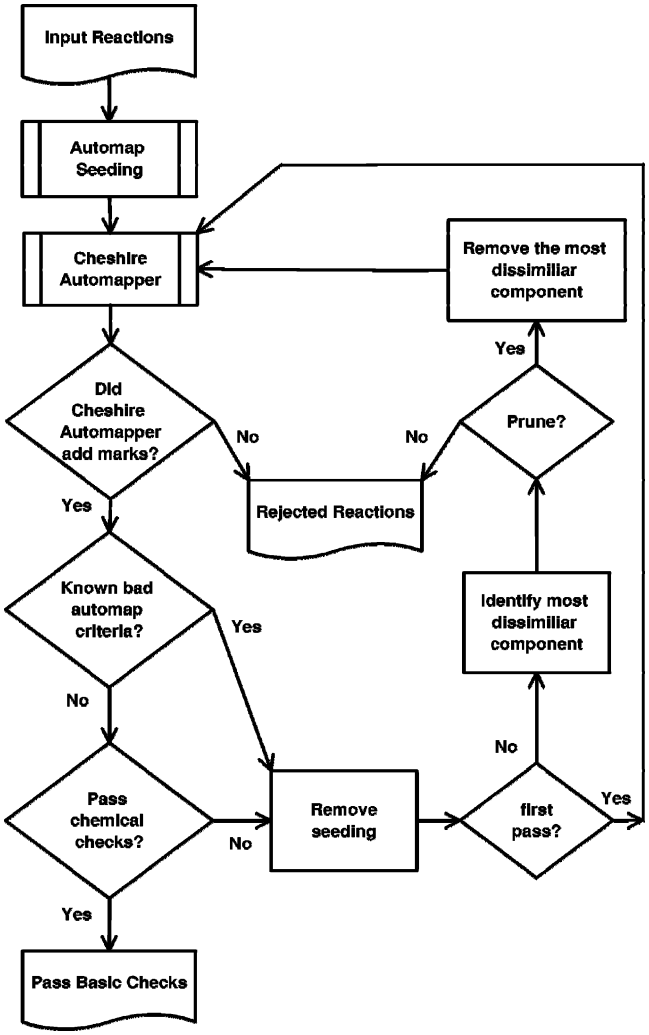

Figure 3. Automapping and basic checks used by VET.

The use of the automappers and Cheshire scripts are detailed in Figure 3.

We begin by using a custom automapper to generate atom−atom map numbers. This automapper is quite conservative in applying mapping numbers; most failures, which are not uncommon, result in no mapping numbers being applied. VET does not explicitly check this automapping for reasonableness.

Instead, the atom−atom map numbers, if present, are used to seed the Cheshire automapper. This seeding is used as a suggestion by the Cheshire automapper and can be changed by the Cheshire automapper. By seeding the Cheshire automapper with results from an independent automapper we increase the fraction of acceptable automappings.

The resulting automapping is examined by VET for plausibility. The first item checked is whether the Cheshire automapper has added any marks to the reaction. Any successful automapping of the reaction by Cheshire will have added either bond marks, reflecting bond changes in the reaction, or inversion/retention marks, reflecting an inversion (or retention) in the configuration of corresponding stereocenters in the reaction. If no marks have been added, the (non)automapping is rejected.

The bulk of VET's plausibility check are checks for features which have been historically linked to automapper failures and checks for chemical plausibility which have been

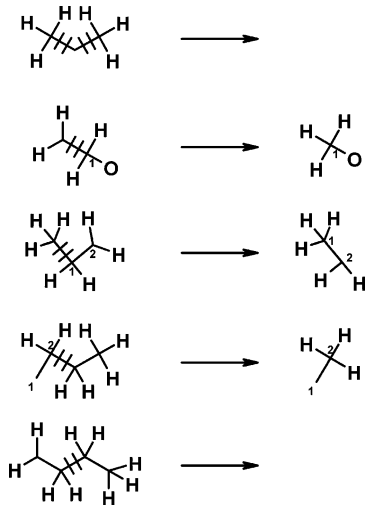

Figure 4. Examples of substructure mapping queries which are replaced by chemical perceptions in VET.

suggested by incorrectly classified reactions.

The checks for "known bad automap criteria" were based on a series of 37 mapping queries which were previously developed to identify Cheshire automappings which had a reasonable probability of being incorrect. These mapping queries are used in production of reaction databases to identify reactions automappings which need human verification. Note that these mapping queries will occasionally select correctly automapped reactions; instead they are chosen because they most often find incorrectly automapped reactions.

Cheshire can also map these queries, but our experience is that replacing simple mapping with perception of the chemical environment leads to increased performance, especially as the number of queries increases. Preliminary perceptions which collect all MAKE/BREAK bonds, all MAKE/BREAK bonds between carbons, all methyl groups, all methylene groups, and all aromatic bonds are used in combination with basic Cheshire chemistry perceptions to build more complex perceptions which replace mapping queries such as those in Figure 4. Perceptions equivalent to the mapping queries in Figure 4 can all be constructed starting with the perception of a MAKE/BREAK bond between carbons. The first query is equivalent to a MAKE/BREAK bond between carbons, with a second alpha to it, having two methyl groups as endpoints of the two bonds and being a reactant. The second query is equivalent to a MAKE/BREAK bond between carbons, with one of the carbon endpoints being a methylene group and the other either a methyl or a methylene group. Additionally, one of the methylene carbons is alpha to an oxygen, and that carbon goes from being a methylene reactant to a methyl product. Heavy use is made of the Cheshire intersection, union and subtraction collection operators as well as the Cheshire Alpha() method, which collects atoms and bonds adjacent to the seed collection. Related perceptions can be placed as nodes in a decision tree, further enhancing performance.

The "chemical checks" include a number of perceptions coded in Cheshire which focus on reaction plausibility. These checks have been constructed in a largely heuristic fashion, with rules constructed to trap incorrect reactions that otherwise would have been accepted. Features which result

in rejection of an automapping include such things as (a) finding aromatic rings which have their substituents change ring location, (b) finding reactions which simultaneously create and destroy triple bonds, and (c) finding reactions which involve insertion into aliphatic carbon−carbon bonds.

As before, these checks are implemented as perceptions in Cheshire and build off of some of the perceptions already carried out as part of the basic checks.

Failure to pass the checks in this part of the program triggers additional passes through the Cheshire automapper. On the first pass the seed map numbers are removed, and the automapping/checking sequence is repeated. Failure in this second pass triggers a third pass in an attempt to produce an acceptable automapping.

The first thing done in the third pass is to identify and remove, or "prune", the most dissimilar reaction component, as long as this can be done without removing the last reactant or product in the reaction. Similarity of reaction components was calculated using Tanimoto coefficients, evaluated using the MDL 960-bit keyset.[12] Following this component removal the automapping/checking sequence is repeated. This continues either until no further components can be removed, in which case the reaction is rejected, or until an acceptable automapping of the simplified reaction is obtained. In this case, the automapping of the simplified reaction is used to seed the Cheshire automapping of the entire reaction.

Automappings which are judged acceptable pass on to final checks of the reaction components.

Final Checks. A final set of checks focus on the presence and type of unmapped atoms in the reaction. These unmapped atoms can result from conventions in reaction representation. Alternatively they can result from misidentified reaction components. More problematically, they could also be indicative of the automapping of a garbled input. Because of this last case, VET assigns large penalties for unmapped atoms, especially product atoms.

We need to support the general practice of suppressing certain classes of reaction components. For example, leaving groups are often not included as products, and Elsevier MDL typically suppresses components which contribute less than 2 carbon atoms to a product. The first of these practices leads to unmapped reactant atoms, the second to unmapped product atoms.

We also want to correct for mistakes in assignment of component roles to REACTANT, PRODUCT, SOLVENT, REAGENT, and CATALYST.

Our component checks start by detecting components which are totally unmapped; these are presumed to be incorrectly included in the reaction and are demoted to an UNKNOWN reaction role.

Next the fraction of unmapped atoms is calculated. If it is below a threshold value (presently 50%), then the reaction is rejected.

Next, we carry out perceptions on unmapped atoms to determine if they are either part of an acceptable suppressed reaction component or are part of a potentially mislabeled reaction component. If the unmapped atoms correspond to atoms in a SOLVENT, REAGENT, or CATALYST component, that component is promoted to either REACTANT or PRODUCT status and the automapping/basic check phase of the algorithm is repeated.

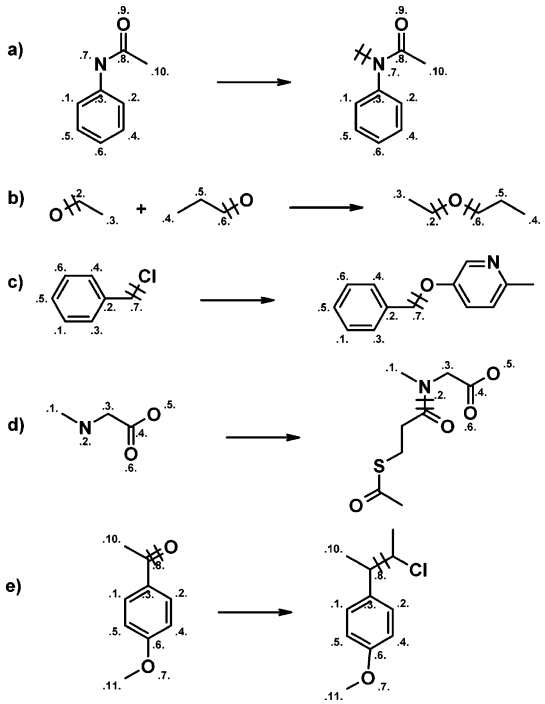

Figure 5. Acceptable unmapped atoms: (a) an unmapped methyl group and (b) an unmapped oxygen/sulfur in a (thio)ester/ether formation/destruction. Unacceptable unmapped atoms: (c) the methylpyridine fragment is unmapped and on the heteroatom side of a C−O bond, (d) the acetylsulfanylpropyl ester fragment is unmapped and on the carbon side of a C−N bond, and (e) a C−C bond, with an unmapped C-atom.

Finally, we look at the remaining unmapped atoms and classify the reactions as either accepted or rejected. Specifically, as detailed in Figure 5: (a) Unmapped groups containing a single carbon atom are acceptable. (b) Unmapped oxygen and sulfur atoms involved in (thio)ester/ether formation/destruction are acceptable. (c) If the reaction makes a C-heteroatom bond and the unlabeled atoms are on the heteroatom-containing moiety, then VET differentiates based on where the unmapped atoms are. VET warns but accepts these if they are both on the reactant side as well as being part of an ester/ether-type reaction. Otherwise they are rejected. (d) If the reaction makes a C-heteroatom bond and the unlabeled atoms are on the C-containing moiety, then VET again differentiates based on where the unmapped atoms are. VET warns but accepts these if they are on the reactant side. Otherwise they are rejected. (e) If the reaction makes a C−C bond and the unlabeled atoms come from either side, then VET will reject them.

PERFORMANCE

As part of VET's development we have collected statistics on both its false positive (accepting incorrectly excerpted reactions) and false negative (rejecting correctly excerpted reactions) rates. Based on human analysis of a sample of approximately 500 input reactions we find that, without correcting for misidentified reaction components, reactions accepted by VET are ∼97% correct, with only ∼3% false positives. Further results from the VET processing of an existing database of 5000 reactions indicated that, again without correcting for misidentified reaction components, VET's rate of rejection of correct reactions is ∼10%.

These results show that there is still room for improvement. However, we note that the false positive rate is comparable to rates expected for humans. Also, the rate of false negatives impacts the size and diversity of the database but not the quality of the entries.

CONCLUSIONS

VET is presently part of a system used in the production of chemical reaction databases by Elsevier MDL. It is part of the automated system which is processing older patent data, a system which is responsible for ~400 000 of the reactions in the patent chemistry database.[9]

There are a number of improvements to the basic VET algorithm which are being examined. They include the following: (a) further refinement of checks for misclassified reaction components. (b) The use of Cheshire-based perceptions to improve the success of automapping. The basic algorithm already does this to a limited degree when it removes dissimilar components from a reaction. Cheshire scripts which help automap reactions with multiple paths and inversion/retention of stereocenters have been developed. (c) The use of MDL reaction keys (which are an extension of the more familiar MDL molecule keys) has been examined. Methods for identification of known reactions with high confidence have been developed. Future work will use these methods to prescreen automapping results for validity.

ACKNOWLEDGMENT

We would like to thank Drs. Stefan Roller, Alexander Lawson, and Bernhard Roth for helpful discussions and Dr. Rong Chen for careful reading of the paper.

REFERENCES AND NOTES

(1) Lynch, M. F.; Willett, P. The Automatic Detection of Chemical Reaction Sites. *J. Chem. Inf. Comput. Sci.* **1978**, *18*, 154−159.

(2) Jochum, C.; Gasteiger, J.; Ugi, I. The Principle of Minimum Chemical Distance (PMCD). *Angew. Chem., Int. Ed. Engl.* **1980**, *19*, 495−505.

(3) McGregor, J. J.; Willett, P. Use of a Maximal Common Subgraph Algorithm in the Automatic Identification of the Ostensible Bond Changes Occurring in Chemical Reactions. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 137−140.

(4) Moock, T. E.; Nourse, J. G.; Grier, D.; Hounshell, W. D. The Implementation of Atom-Atom Mapping and Related Features in the Reaction Access System (REACCS) In *Chemical Structures*; Warr, W., Ed.; Springer-Verlag: Berlin, 1988; pp 303−313.

(5) Chen, L.; Nourse, J. G.; Christie, B. D.; Leland, B. A.; Grier, D. L. Over 20 Years of Chemical Structure Access Systems from MDL: Evolution of Reaction Substructure Methods. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1296−1310.

(6) Moock, T. E.; Grier, D. L.; Hounshell, W. D.; Grethe, G.; Cronin, K.; Nourse, J. G.; Theodosiou, J. Similarity Searching in the Organic Reaction Domain. *Tetrahedron Comput. Methodol.* **1988**, *1*, 117−128.

(7) Chen, L. Reaction Classification and Knowledge Acquisition. In *Handbook of Chemoinformatics*; Gasteiger, J., Ed.; Wiley-Vch: Wein Heim, 2003; Vol. 1, pp 348−388.

(8) Cheshire, Elsevier MDL, San Leandro, CA, U.S.A.

(9) Patent Chemistry Database, Elsevier MDL, San Leandro, CA, U.S.A.

(10) (a) 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. (b) MDL CtFile Formats, Elsevier MDL, San Leandro, CA, U.S.A.

(11) Gushurst, A. J.; Nourse, J. G.; Hounshell, W. D.; Leland, B. A.; Raich, D. G. The substance module: the representation, storage, and searching of complex structures. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 447−454.

(12) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimiztion of MDLKeys for Use in Drug Discovery. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1273−1280.

CI050390E