Automated Generation of Kinetic Chemical Mechanisms Using Rewriting

Olivier Bournez[1], Guy-Marie Côme[2], Valérie Conraud[2], Hélène Kirchner[1], and Liliana Ibănescu[1]*

[1] LORIA & INRIA Lorraine
615, rue du Jardin Botanique, BP 101
F-54602 Villers-lès-Nancy Cedex, France
{Olivier.Bournez,Helene.Kirchner,Mariana-Liliana.Ibanescu}@loria.fr
[2] Département de Chimie Physique des Réactions (DCPR), INPL-ENSIC
1, rue Grandville, BP 451
F-54001, Nancy, France
{Guy-Marie.Come, Valerie.Conraud}@ensic.inpl-nancy.fr

Abstract. Several software systems have been developed recently for the automated generation of combustion reactions kinetic mechanisms using different representations of species and reactions and different generation algorithms. In parallel, several software systems based on rewriting have been developed for the easy modeling and prototyping of systems using rules controlled by strategies. This paper presents our current experience in using the rewrite system ELAN for prototyping the automatic generation of the combustion reactions mechanisms previously implemented in the EXGAS kinetic mechanism generator system. We describe how to express in ELAN acyclic and cyclic molecules, reactants, elementary reactions and the primary mechanism for acyclic species. Examples and generated outputs are given.

1 Introduction

Combustion reactions will still be the main source of energy in the 21st century. Understanding the fundamental mechanisms of these reactions is highly desirable for the optimal design and operation of efficient, safe and clean engines, burners, incinerators (see for e.g. [5]). Due to the complexity of mechanisms, several software systems have been developed for automatic generation of combustion reactions mechanisms using different representations of species and reactions, and different generation algorithms. Even if only simple molecules are used as reactants, the number of elementary reactions generated is very often big: for example 3662 reactions involving 470 species are generated by the software system EXGAS for explaining the combustion mechanism of n-hexane [7]. Traditional techniques for automatic generation of mechanism are limited by the computational power of computers, and by the difficulty to express chemists expertise into computational concepts.

* Work supported by Peugeot Citroën Automobiles.

P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2659, pp. 367–376, 2003.

Term and graph rewriting has been developed since the last thirty years, leading to a deep and solid corpus of knowledge about the rewrite relation induced by a set of equations (see for e.g. [2]). More recently, rule based languages focused on the use of rewriting as a modeling tool, which results in making the out-coming specification executable in a very efficient way (see for e.g. [8]).

The aim of our study is to use rewrite systems for automatic generation mechanisms. We plan to benefit from the elegance and expressiveness of rewriting as a computational paradigm implemented in the ELAN system [1,3] developed by the team of computer scientists at LORIA, and from the chemical expertise from the team of chemists at DCPR that developed EXGAS [5,15].

The paper is organized as follows: in Sect. 2 we describe the problem of generation of kinetic mechanisms, in Sect. 3 we introduce some notions about rewrite systems and in Sect. 4 we explain how to code in ELAN chemical objects like acyclic and cyclic molecules (in Sect. 4.1), reactants (in Sect. 4.2), elementary reactions (in Sect. 4.3) and the primary mechanism for acyclic species (in Sect. 4.5).

The results obtained so far and the positive feedback from chemists encourage us to continue our study, to set-up a prototype and to extend our technique to cyclic molecules.

2 Automatic Generation of Combustion Reactions Mechanisms

Combustion reactions take place in engines, burners and industrial chemical reactors to produce mechanical or thermal energy, and also to incinerate pollutants or to manufacture chemical substances. In order to understand the fundamental mechanism of combustion reactions in particular and gas-phase thermal reactions in general, different models are used for numerical simulations of the phenomena.

A methodology for the elaboration of fundamentals models for the gas-phase thermal reactions is given in [5] and the three main phases are described in Fig. 1:

- The elaboration of the initial reaction model consists of generating a mechanism for the reaction, by hand or using a computer program, of estimating the thermodynamic data of molecules and free radicals involved in the mechanism and the kinetic data of the elementary reactions constituting the mechanism.
- The numerical resolution of the balance equations of the laboratory reactor; the input data are the initial model of reaction, the operating conditions, as well as, possibly, transport data for the species.
- The validation of the reaction model by comparison of the results of simulations with experimental results and adjustment where necessary of certain thermodynamic, kinetic or transport parameters.

The final reaction model generated by this methodology can be incorporated into a calculation code of a reactor or a computer program of reacting fluid dynamics for simulations.

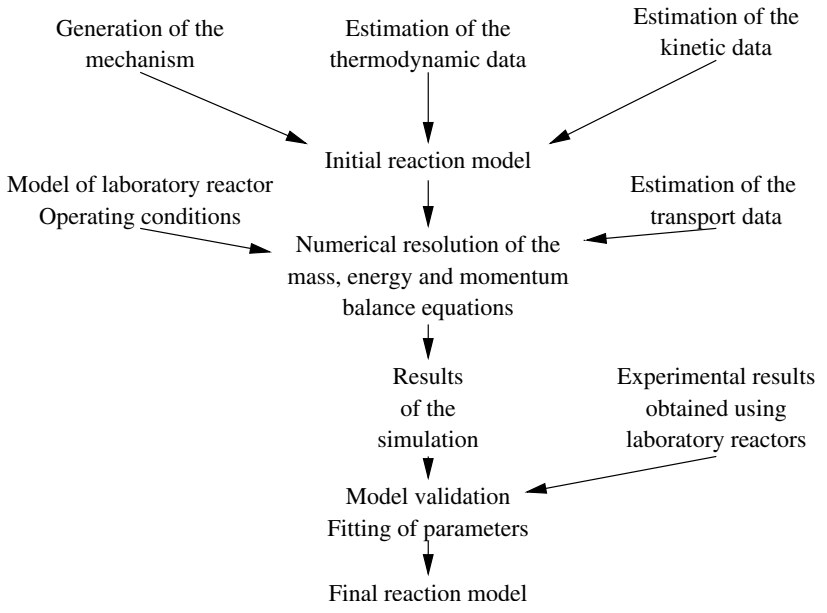

Fig. 1. Elaboration and validation of a model of a gas-phase thermal reaction

The generation of combustion mechanisms for a mixture of organic compounds in a large temperature field requires to consider several hundred chemical species and several thousands of elementary reactions. Therefore using computers is unavoidable. The generation of reactions mechanisms assisted by the computer has been rendered possible due to experimental and theoretical knowledge accumulated over the last three decades about the thermodynamic properties of molecules and free radicals, about elementary reactions and about the morphology of the reaction mechanism themselves (see for e.g. [5]).

A number of software systems have been developed for automatic generation of detailed mechanisms using both the expertise of researchers in thermodynamics and statistical mechanics, chemical kinetics, transport phenomena and theory of reactors and the expertise of computer scientists. A non exhaustive list of these software systems is the following: MAMOX [13], NetGen [14], EXGAS [5, 15], COMGEN [12].

Most of these systems are implemented using traditional imperative programming, using rather ad hoc data-structures and procedures for the representation of molecules and for the generation of the mechanisms. In particular, none of them benefits from flexibility and ease of modification of the rules that are used for the generation of mechanisms, even if their practical use requires to often modify the rules, according to chemist expertise or according to new experimental data.

3 Rewrite Systems

Rule-based programming is currently experiencing a renewed period of growth with the emergency of new concepts and systems that allow one to better understand and better use it. On the theoretical side, after the in-depth study of rewriting concepts during the eighties, the last decade saw the emergence of the general concepts of rewriting logic and of rewriting calculus. On the practical side, systems like ASF+SDF [9], Maude [4], Cafe-OBJ [6], or ELAN [3] and also commercial products like Ilog Rules, have shown that the concept of rule can be of major interest as a programming tool.

The ELAN system [1,3] provides an environment for specifying and prototyping deduction systems in a language based on rules controlled by strategies. The first class objects manipulated are rewriting rules, which may be used to model chemical reactions.

In ELAN, a program is a set of labeled conditional rewrite rules with local affectations

$$\ell : l \Rightarrow r \text{ if } c \text{ where } w$$

Informally, rewriting a ground term t consists of selecting a rule whose left-hand side (also called pattern) matches the current term (t), or a sub-term ($t_{|\omega}$), computing a substitution σ that gives the instantiation of rule variables ($l\sigma = t_{|\omega}$), and if the instantiated condition c is satisfied ($c\sigma$ reduces to $true$), applying substitution σ enriched by local affectation w to the right-hand side to build the reduced term.

In general, the normalization of a term may not terminate, or may terminate with different results corresponding to different selected rules, selected sub-terms or non-unicity of the substitution. So evaluation by rewriting is essentially non-deterministic and backtracking may be needed to generate all results. One of the main originalities of the ELAN language is to provide strategies as first class objects of the language. This allows the programmer to specify in a precise and natural way the control on the rule applications.

The full ELAN system includes a preprocessor, an interpreter, a compiler, and standard libraries available through the ELAN web page [1]. From the specific techniques developed for compiling strategy controlled rewrite systems [10,11], the ELAN compiler is able to generate code that applies up to 15 millions rewrite rules per second on typical examples where no non-determinism is involved and typically between 100 000 and one million controlled rewrite per second in presence of associative-commutative operators and non-determinism. We think that this may help to handle the combinatorial explosion of the number of reactions modeling a kinetic mechanism.

Note that among the possibilities offered by the ELAN system that we will use in this paper, we can notice that the ELAN system offers the power of mix-fix signatures and matching, the possibility of defining and using associative-commutative operators.

4 Chemical Objects in ELAN

4.1 Molecule Representation

Molecules are represented in a simplified form as labeled graphs, describing the composition and the constitution of a molecule: vertices correspond to atoms, edges represent covalent bonds between atoms; bond lengths or bond angles are not taken into account.

In order to represent molecules in ELAN we use a *term* representation inspired from chemical linear SMILES notation [16] that code molecular graphs as trees:

1. Molecules are represented as hydrogen-suppressed molecular graphs.
2. If the hydrogen-suppressed molecular graph has cycles, we transform it into a tree applying the following rule to every cycle: choose one fresh digit and one single or aromatic bond of the cycle, break the bond and label the 2 atoms with the same digit.
3. Choose a root of the tree, and represent it like a concatenation of the root and the list of its sons.

For example, the term `C 1 (C) C C 1` represents methylcyclopropane and the term `C C (= O)O` represents acetic acid: see Fig. 2.

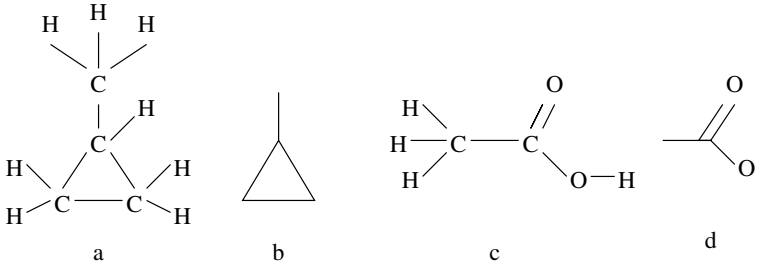

Fig. 2. Representations of molecules: a. molecular graph of methylcyclopropane; b. corresponding hydrogen-suppressed molecular graph; c. molecular graph of Acetic acid; d. corresponding hydrogen-suppressed molecular graph

In Fig. 3 we give the mix-fix signature in ELAN of our notation.

4.2 Reactant Representation

Reactants are represented in ELAN using an associative-commutative operator `+`. For example, `C C C C + O=O + C C (= O)O` is a term that represents a mixture of n-butane, oxygen and acetic acid.

```
C, c, O, o, H   : symbol;     /* Atom specification */
@ @     : (symbol int) symbol;/* Labels for cycle closure specification*/
 -, =, #, ':'   : link;       /* Link specification */
e       : radical;
@       : (symbol) radical;   /* Molecules and radicals specification */
@ @     : (symbol radical_list) radical;
@       : (radical) radical_list;    /* List of radicals specification */
@ @     : (link radical) radical_list;
(@)     : (radical_list) radical_list;
(@) @   : (radical_list radical_list) radical_list (AC);
```

Fig. 3. ELAN signature of our SMILES like notation for molecules

4.3 Elementary Reactions in ELAN

The ten main classes of generic reactions in EXGAS are: 1) unimolecular initiation (ui); 2) bimolecular initiation (bi); 3) addition of free radicals to oxygen (ad); 4) isomerisation of free radicals (is); 5) unimolecular decomposition of free radicals by beta-scission (bs); 6) unimolecular decomposition of hydroperoxyalkyl free radicals to cyclic ethers (cy); 7) oxidation of free radicals (ox); 8) metathesis (me); 9) combination of free radicals (co); 10) disproportionation of free radicals (di).

Generic reactions like addition of free radicals to oxygen, oxidation of free radicals, combination of free radicals and disproportionation of free radicals are expressed in a direct way in ELAN: the generic reaction is expressed into a term transformation which is described using ELAN code into rewrite rules.

For example, the generic reaction for addition of free radicals to oxygen is the following

$$O{=}O + \bullet R \longrightarrow \bullet OOR$$

It corresponds to the graph transformation given in Fig. 4 and is coded directly

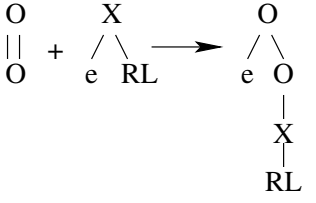

Fig. 4. Term transformation coding addition of free radicals to oxygen

by the following ELAN rewriting rule:

```
[ad] O=O + X (e) RL => O (e) O X RL end
```

where `X` is a variable of sort `symbol` and `RL` is a variable of sort `radical_list`.

The ELAN system generates the following output as the result for addition of butyl to oxygen:

```
Addition of C(e) C C C to O=O
 [ad] O=O + C(e)C C C  ----->  O(e)O C C C C
```

4.4 Generating All Visions

The generic reaction for unimolecular initiation is the following

$$\text{A–B} \longrightarrow \bullet\text{A} + \bullet\text{B}$$

Breaking the `C`–`C` bonds during unimolecular initiation corresponds to the term transformation given in Fig. 5 and can be implemented by the following ELAN

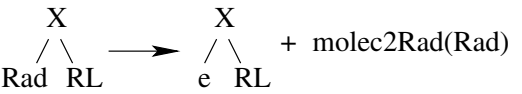

Fig. 5. Term transformation coding unimolecular initiation

rewriting rule, a *named rule*

```
[ui] X (Rad) RL => X(e) RL + molec2Rad(Rad) end
```

where `X` is a variable of sort `symbol`, `Rad` is a variable of sort `radical` and `RL` is a variable of sort `radical_list`. The operator `molec2rad( )` transforms its argument into a radical and is defined by two ELAN rewriting rules, *no-named* rules:

```
[] molec2rad(X Rad)       => X (e) Rad  end
[] molec2rad(X (Rad) RL) => X (e) (Rad) RL  end
```

Note that the power of the associative-commutative matching allows us to give one generic ELAN rewriting rule for the unimolecular initiation and it will be applied to all sub-terms of root `X`.

The ELAN system applies a named rule to a term by matching the left-hand side of the rule to the term. In order to obtain all the results of the unimolecular initiation for the molecule `X (Rad) RL`, the rewriting rule `[ui]` has to be applied everywhere inside the term. To do this we suggest the following technique:

1. We define the *vision of a molecule* to be a representation of a molecule as a tree, given by the choice of the root (see Fig. 6).
2. We provide an operator `AllVis` that generates all the visions of a molecule by choosing every node of the tree representing the molecule to be the root.

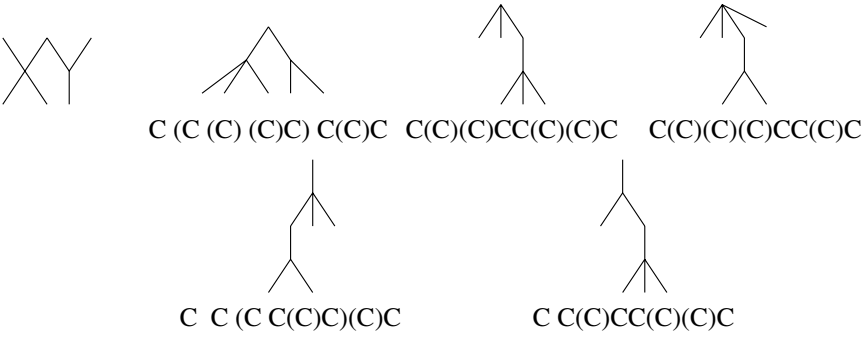

Fig. 6. The set of all distinct visions for ISO-octane

3. We apply the generic ELAN rewriting rule for unimolecular initiation to every vision of the molecule given by the `AllVis` operator.

The ELAN system generates the four first reactions from Fig. 7 as the result for the unimolecular initiation of ISO-octane by breaking the `C`–`C` bonds.

Equality Test for Two Molecules. Equality of two molecules M_1 and M_2, represented in the ELAN signature, means that M_1 and M_2 are visions of the same hydrogen-suppressed molecular graph. The `AllVis` operator is used for the equality test:

$$\texttt{is\_eq}(M_1, M_2) \texttt{ => true if } M_1 \in \texttt{AllVis}(M_2)$$

4.5 Primary Mechanism in ELAN

The primary mechanism in EXGAS is the following:

1. initiation reactions (ui and bi, reaction patterns 1 and 2 from Sect. 4.3) are applied to initial reactants (a mixture of molecules) and free radicals are generated;
2. the set of generic propagation reactions (reaction patterns 3—8 from Sect. 4.3) are applied to the free radicals until no new radical is generated;
3. the termination reactions (reaction patterns 9 and 10) are applied to free radicals and generate molecules.

In ELAN the primary mechanism is defined in a natural way as the concatenation of three strategies corresponding to each phase, `tryInit` for the initiation phase, `tryPropag` for the propagation phase and `tryTermin` for the termination phase:

```
[] mec_prim     => tryInit; tryPropag; tryTermin end
```

The user defined strategies `tryInit` and `tryTermin` are expressed using the ELAN choice strategy operator **dk** applied to the strategies (the ELAN rewriting rule) defining the generic reactions. The **dk** operator (*dont know choose*) takes all strategies given as arguments and returns, for each of them the set of all its results.

For example the following ELAN rule expresses a strategy that returns all the results of the unimolecular and bimolecular initiation:

```
[] tryInit      =>  dk(ui, bi) end
```

The output generated by ELAN after the initiation phase was applied to ISO-octane is given in Fig. 7.

```
Initiations for  C(C(C)C)C(C)(C)C
 [ui] C C(C)C C(C)(C)C -----> C e + C(C)(C C(C)(C)C)e
 [ui] C C(C)(C)C C(C)C -----> C e + C(C)(C)(C C(C)C)e
 [ui] C(C(C)C)C(C)(C)C -----> C(C(C)C)e + C(C)(C)(C)e
 [ui] C(C(C)C)C(C)(C)C -----> C(C(C)(C)C)e + C(C)(C)e
 [bi] O=O + C(C)(C)C C(C)(C)C -----> C(C)(C)(C C(C)(C)C)e + O(e)O
 [bi] O=O + C C(C)C C(C)(C)C -----> C(C(C)C C(C)(C)C)e + O(e)O
 [bi] O=O + C C(C)(C)C C(C)C -----> C(C(C)(C)C C(C)C)e + O(e)O
 [bi] O=O + C(C(C)C)C(C)(C)C -----> C(C(C)C)(C(C)(C)C)e + O(e)O
```

Fig. 7. Initiation reactions of ISO-octane combustion in ELAN

Strategy `tryPropag` is defined as the iteration of one step of propagation using the ELAN strategy iterator **repeat***.

```
[] tryPropag => repeat*(propagOne) end
```

Strategy **repeat** iterates the strategy until it fails and returns the terms resulting from the last unfailing call of the strategy.

Strategy `propagOne` is defined in a similar way as `tryInit` using a **dk** operator applied to the generic reactions of the propagation phase.

5 Conclusions and Further Work

We described in this paper how to express in ELAN: acyclic and cyclic molecules, reactants, elementary reactions and the primary mechanism for acyclic species. This work is in a preliminary phase, since not all the functionalities of the EXGAS system have been implemented using the ELAN system. However, we think that at this time we have proved the feasibility of this approach.

The results obtained so far and the positive feedback from chemists encourage us to continue our study, to set-up a prototype and to extend it to cyclic molecules, which are not yet fully supported by any of the mentioned automatic kinetic mechanism generator systems.

References

1. Elan web site. `http://elan.loria.fr`.
2. Franz Baader and Tobias Nipkow. *Term Rewriting and all That*. Cambridge University Press, 1998.
3. Peter Borovanský, Hélène Kirchner, Pierre-Etienne Moreau, and Christophe Ringeissen. An overview of elan. In Claude Kirchner and Hélène Kirchner, editors, *Electronic Notes in Theoretical Computer Science*, volume 15. Elsevier Science Publishers, 2000.
4. M. Clavel, F. Duran, S. Eker, P. Lincoln, N. Marti-Oliet, J. Meseguer, and J.F. Quesada. Towards maude 2.0. In Kokichi Futatsugi, editor, *Electronic Notes in Theoretical Computer Science*, volume 36. Elsevier Science Publishers, 2001.
5. Guy-Marie Côme. *Gas-Phase Thermal Reactions. Chemical Engineering Kinetics.* Kluwer Academic Publishers, 2001.
6. K. Futatsugi and A. Nakagawa. An overview of CAFE specification environment – an algebraic approach for creating, verifying, and maintaining formal specifications over networks. In *Proceedings of the 1st IEEE Int. Conference on Formal Engineering Methods*, 1997.
7. Pierre-Alexandre Glaude, Frédérique Battin-Leclerc, René Fournet, Valérie Warth, Guy-Marie Côme, and Gérard Scacchi. Construction and simplification model for the oxidation of alkanes. *Combustion and Flame*, 122:451–462, 2000.
8. Hélène Kirchner and Pierre-Etienne Moreau. Promoting rewriting to a programming language: A compiler for non-deterministic rewrite programs in associative-commutative theories. *Journal of Functional Programming*, 11(2):207–251, 2001.
9. Paul Klint. A meta-environment for generating programming environments. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2(2):176–201, 1993.
10. Pierre-Etienne Moreau. Rem (reduce elan machine): Core of the new elan compiler. In Leo Bachmair, editor, *Rewriting Techniques and Applications, 11th International Conference, RTA 2000, Norwich, UK, July 10-12, 2000, Proceedings*, volume 1833 of *Lecture Notes in Computer Science*, pages 265–269. Springer, 2000.
11. Pierre-Etienne Moreau and Hélène Kirchner. A compiler for rewrite programs in associative-commutative theories. In Catuscia Palamidessi, Hugh Glaser, and Karl Meinke, editors, *Principles of Declarative Programming, 10th International Symposium, PLILP'98 Held Jointly with the 7th International Conference, ALP'98, Pisa, Italy, September 16-18, 1998, Proceedings*, volume 1490 of *Lecture Notes in Computer Science*, pages 230–249. Springer, 1998.
12. Artur Rakiewicz and Thanh N. Truong. Application of chemical graph theory for automated mechanism generation. *accepted for publication*, 2002.
13. E. Ranzi, T. Faravelli, P. Gaffuri, and A. Sogaro. Low-temperature combustion: automatic generation of primary oxydation reaction and lumping procedures. *Combustion and Flame*, 102:179, 1995.
14. Roberta G. Susnow, Anthony M. Dean, William H. Green, P.Peczak, and Linda J. Broadbelt. Rate-based construction of kinetic models for complex systems. *The Journal of Physical Chemistry A*, 101:3731–3740, 1997.
15. Valérie Warth, Frédérique Battin-Leclerc, René Fournet, Pierre-Alexandre Glaude, Guy-Marie Côme, and Gérard Scacchi. Computer based generation of reaction mechanisms for gas-phase oxidation. *Computers and Chemistry*, 24:541–560, 2000.
16. David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Science*, 28:31–36, 1988.