



HAL
open science

ATOMOL : un système d'aide au suivi des travaux dirigés de chimie en premier cycle universitaire

Dominique Fischer, Michèle Kirch, Amedeo Napoli

► To cite this version:

Dominique Fischer, Michèle Kirch, Amedeo Napoli. ATOMOL : un système d'aide au suivi des travaux dirigés de chimie en premier cycle universitaire. Sciences et Techniques Educatives, 1995, 2 (1), pp.61-102. edutice-00001454

HAL Id: edutice-00001454

<https://edutice.hal.science/edutice-00001454v1>

Submitted on 26 Oct 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ATOMOL : un système d'aide au suivi des travaux dirigés de chimie en premier cycle universitaire

Dominique Fischer^{*1} — Michèle Kirch^{**} — Amedeo Napoli^{***}

** CAFIE, Centre d'Autoformation et d'Ingénierie Educative
Université Louis Pasteur, Institut Le Bel, 4 rue Blaise Pascal,
67000 Strasbourg Cedex, France*

***GERSULP, Groupe d'Études et de Recherches sur les Sciences
Université Louis Pasteur, Institut Le Bel, 4 rue Blaise Pascal,
67000 Strasbourg Cedex, France
E-mail : mKirch@chimie.u-strasbg.fr*

****CRIN CNRS -- INRIA Lorraine
BP 239 -- 54506 Vandoeuvre-Les-Nancy Cedex -- France
E-mail : napoli@loria.fr*

RESUME. Dans cet article, nous présentons ATOMOL, un logiciel d'enseignement assisté par ordinateur destiné aux étudiants en première année de DEUG. L'architecture du système est classique, mais sa conception est originale : elle repose entièrement sur un formalisme objet, aussi bien pour la représentation des connaissances chimiques que pour la représentation générale des exercices et des problèmes. Ainsi, le système intègre un niveau spécifique à la chimie, et un niveau générique réutilisable dans d'autres contextes. Après une brève introduction sur le rôle de l'ordinateur dans l'enseignement de la chimie, l'article contient une présentation détaillée de ATOMOL, et se termine par une discussion sur les qualités informatiques et l'intérêt pédagogique du logiciel.

ABSTRACT. In this paper, we present ATOMOL, a tutoring system, especially built for students taking a first course in chemistry. The architecture of the system is rather classical, but the design of the system is original: knowledge representation relies on an object-oriented basis, both for basic chemical knowledge and for generic problem description and problem solving. Thus, the system has two main components, a specific component for chemical knowledge, and a generic reusable component for tutoring. The paper includes first a presentation of tutoring systems for chemistry, then a detailed description of ATOMOL, and lastly, a discussion about the capabilities of ATOMOL regarding programming and tutoring.

MOTS CLES : enseignement assisté par ordinateur, représentation des connaissances, chimie, représentation à objets.

KEYWORDS : computer aided instruction, tutoring system, knowledge representation, chemistry, object-based representation.

¹ La première version du logiciel ATOMOL a été développée au Laboratoire des Modèles Informatiques Appliqués à la Synthèse de l'Université Louis Pasteur à Strasbourg, dans le cadre de la thèse d'Université préparée par le premier auteur.

1. Introduction

Traditionnellement, l'enseignement supérieur de chimie est composé de cours magistraux, de séances de travaux dirigés illustrant les cours et de séances de travaux pratiques en laboratoire. Actuellement, c'est dans le laboratoire que l'ordinateur a trouvé le plus grand nombre d'applications, dans toutes les phases de l'expérimentation :

- préparation d'une manipulation : rappels de cours et des consignes opératoires par l'intermédiaire de logiciels d'enseignement assisté par ordinateur [PLL 93], préparation à l'utilisation du matériel à l'aide de logiciels de simulation [MSA 80][HOL 84],
- pendant le déroulement de la manipulation : couplage d'appareils de chimie avec un ordinateur pour le pilotage de la manipulation, l'acquisition et le traitement des données [AD 93],
- en fin de manipulation : interprétation des résultats [SET 87].

Dans les cours de chimie, l'ordinateur peut être un moyen original de visualisation des modèles et des phénomènes chimiques, par exemple à l'aide de modèles moléculaires informatiques [AAF+ 91] ou de logiciels d'étude des structures cristallines [DLL 89]. En ce qui concerne les travaux dirigés, il existe un certain nombre de logiciels qui peuvent apporter un complément à l'enseignement traditionnel. Des travaux intéressants ont été réalisés, par exemple pour l'aide à l'apprentissage des règles de nomenclature des composés chimiques [NAU 89][CLE 89], pour l'étude de l'aspect structural et géométrique des molécules [BMA 84][DDR 89], ou encore en spectroscopie [CRF 88][RRC+ 88][CRR+ 91]. Plusieurs universités possèdent un centre de ressources informatiques où les étudiants peuvent travailler en autoformation avec ce type de logiciels. C'est le cas de l'université Louis Pasteur de Strasbourg, où le logiciel ATOMOL est en cours de développement.

Notre principal objectif est de donner aux étudiants la possibilité de s'entraîner individuellement et librement sur des concepts de base en chimie, de manière à ce que chaque étudiant puisse progresser à son rythme et travailler conformément à son niveau. Les sujets traités par le logiciel ATOMOL portent sur des notions de chimie élémentaire indispensables au chimiste, comme les modes de représentation électronique des molécules (formules de Lewis), la détermination de la géométrie des molécules et les problèmes de stéréochimie. En particulier, soulignons que ce type de problème n'est pas ou n'est qu'en partie pris en compte par les logiciels d'enseignement assisté par ordinateur en chimie qui existent à l'heure actuelle². La compréhension et l'acquisition de ces notions se font par la pratique, autrement dit par un entraînement suffisant et répétitif ; l'enseignement assisté par ordinateur se prête particulièrement bien à un tel entraînement. Nous avons également tenu à intégrer dans ATOMOL des fonctionnalités réservées à l'enseignant, qui lui

² Le catalogue du C.D.I.E.C (Centre Documentaire Informatique Enseignement Chimie, Université de Nice Sophia Antipolis) répertorie l'ensemble des logiciels d'enseignement assisté par ordinateur disponibles en chimie. Le logiciel VSEPR [DDR 89] qui devait traiter le problème de la géométrie des molécules n'est pas opérationnel.

permettent de suivre les travaux effectués par les étudiants, d'ajouter des données aux tâches proposées par le logiciel et de gérer le déroulement des séances de travail.

La réalisation du logiciel ATOMOL a été l'occasion d'intégrer dans un nouveau type d'application une représentation à objets des molécules conçue initialement pour un système de synthèse assistée par ordinateur [NAP 92][NL 93]. Ainsi, nous avons voulu montrer la généralité et le caractère évolutif des représentations à objets. Pour la conception du logiciel, deux axes de travail ont été suivis en parallèle : la représentation informatique de l'expertise chimique et l'intégration de cette représentation dans un environnement dédié à l'enseignement. Ce travail a fait l'objet d'une thèse qui est la référence de base pour tout cet article [FIS 93]. En ce qui concerne la chimie, aucune connaissance spécialisée n'est requise, la référence générale étant [ARN 83].

Le cœur de l'article se compose de cinq grandes parties. Dans la première est présentée l'architecture du logiciel d'enseignement assisté par ordinateur ATOMOL. Les bases de l'implantation du logiciel sont examinées dans la deuxième partie, et les raisons d'utilisation d'un langage à objets y sont précisées. Les trois parties suivantes découlent directement de l'architecture du système, puisqu'elles décrivent respectivement la connaissance chimique de base (le module expert), les caractéristiques de l'enseignement associées à ATOMOL (le module d'enseignement) et enfin les méthodes de suivi des étudiants (l'historique de l'étudiant). L'article se termine par une discussion sur les qualités pédagogiques et informatiques du logiciel ATOMOL. Les nombreuses présentations d'objets génériques décrivant des concepts chimiques et pédagogiques font également de cet article un exemple détaillé de conception de représentation à objets.

2. Le logiciel ATOMOL

2.1. Architecture du logiciel ATOMOL

L'architecture que nous avons adoptée pour réaliser le logiciel ATOMOL est calquée sur celle que l'on retrouve généralement dans les tuteurs intelligents, logiciels d'enseignement assisté par ordinateur qui utilisent des techniques d'intelligence artificielle [NV 88][QUE 91]. Le logiciel comprend donc un module expert, un module d'enseignement, un historique de l'étudiant et une interface (figure 1). Le module expert contient la description des connaissances du domaine à enseigner et les mécanismes de raisonnement sur ces connaissances. La fonction du module d'enseignement est de gérer le déroulement de l'apprentissage. Le troisième module, l'historique de l'étudiant, est un ensemble d'informations concernant l'étudiant, qui sont exploitées par le module d'enseignement pour adapter l'apprentissage à l'étudiant. La communication entre l'étudiant ou l'enseignant et le module d'enseignement s'effectue au travers de la dernière composante, l'interface.

Du point de vue de l'implantation, le logiciel ATOMOL est un système à base de connaissances, la base étant constituée par une représentation à objets des molécules,

programmée en YCHEM³, un langage à objets écrit en LISP [NAP 90] [NAP 92]. Pour construire l'environnement d'enseignement autour de la représentation des molécules, nous avons utilisé naturellement les mêmes langages, LISP et YCHEM, ce qui rend le système totalement homogène. Ainsi, dans ATOMOL, les connaissances chimiques de base mais aussi les éléments propres à l'enseignement sont représentés par des objets génériques qui intègrent données et fonctions. Les caractéristiques générales de ces objets sont explicitées dans le paragraphe suivant.

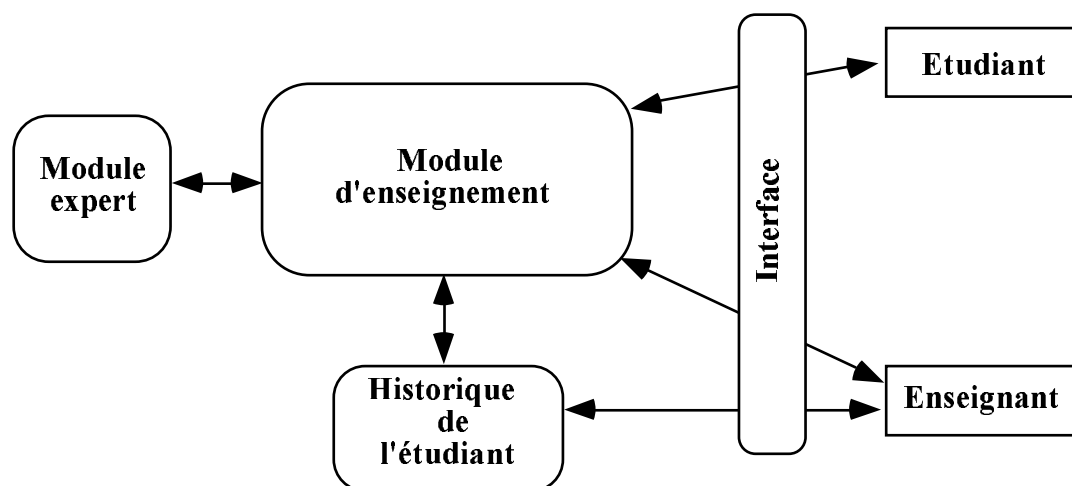


Figure 1. Architecture du logiciel ATOMOL.

2.2. Les objets dans ATOMOL

2.2.1. Pourquoi une représentation à objets ?

Dans ce qui suit, les connaissances qui permettent de décrire et de manipuler une molécule sont représentées par des objets. Dans une représentation à objets, l'objet est l'unité de connaissance et il peut être de deux types, générique ou élémentaire. Un objet générique, appelé aussi classe, regroupe un ensemble d'attributs et un ensemble de méthodes qui forment un tout décrivant un concept. Un objet générique possède un certain nombre de représentants particuliers, encore appelés instances, qui sont les objets élémentaires effectivement manipulés dans les applications. De fait, toute l'information propre à un concept est localisée en une seule unité homogène, même si un objet générique partage un certain nombre de propriétés avec d'autres objets génériques par l'intermédiaire du mécanisme d'héritage. Une représentation à objets évite donc la dispersion des caractéristiques associées à un concept et assure une bonne cohésion des descriptions. Lorsqu'elles sont possédées par un langage de représentation, les qualités d'homogénéité et de cohésion facilitent l'évolution et la maintenance des logiciels créés avec ce langage [MNC+ 89]. Enfin, l'objet est une

³YCHEM est une version réduite de YAFOOL [DUC 88] adaptée aux micro-ordinateurs (Macintosh ou PC).

unité de représentation qui autorise aussi une bonne compréhension et une bonne lisibilité des structures décrites. Par rapport à d'autres structures de données plus classiques utilisées pour représenter un modèle constitutionnel des molécules comme les tables de connexion [AH 75], une représentation à objets possède donc des avantages certains.

2.2.2. Le langage à objets YCHEM

Le langage à objets YCHEM permet de construire et de manipuler des objets génériques appelés *frames* ou *modèles*. Un frame décrit un concept du monde réel, un atome par exemple, et se présente comme une structure à trois niveaux, *frame-propriétés-facettes* [MNC+ 89]. Les propriétés se divisent en *attributs* et *méthodes*. Un attribut décrit une caractéristique du frame. Un attribut est lui-même décrit par une ou plusieurs facettes, qui expriment des modalités descriptives ou comportementales attachées à l'attribut. A chaque facette est associée une "valeur", qui peut être une valeur élémentaire, mais aussi un type, ou un domaine, ou encore un *réflexe* (fonction qui se déclenche lors de l'accès à l'attribut). Une méthode décrit un comportement associé au frame et elle est activée par un *envoi de message*.

La *relation de spécialisation* permet d'organiser les frames en une hiérarchie ou graphe d'héritage où les frames génériques partagent des propriétés avec des frames spécifiques. Dans la hiérarchie, les ascendants d'un frame sont appelés ses *superframes*, et ses descendants, ses *sous-frames*. L'héritage est multiple car un frame peut avoir plusieurs superframes. Il existe deux manières de *spécialiser* un frame. La première consiste à l'enrichir, c'est-à-dire à ajouter de nouvelles propriétés qui n'existent pas dans les superframes. La seconde est le masquage, qui consiste à redéfinir une propriété qui est déjà présente dans un des superframes. Le partage de propriété entre un frame et ses superframes est *dynamique* : les caractéristiques des superframes ne sont consultées que par nécessité. La hiérarchie a une racine, appelée *Objet-Ideal* en YCHEM, qui possède des propriétés d'ordre général (impression d'un frame, visualisation, etc.) héritées par tous les autres frames de la hiérarchie.

A partir d'un frame, on peut fabriquer un grand nombre d'objets élémentaires du même type, qui possèdent les caractéristiques du frame. Ces objets particuliers ont la même structure que le frame et sont appelés les *instances* du frame. Les instances sont des objets terminaux, qui ne peuvent plus être spécialisés.

La figure 2 présente une description simplifiée de la hiérarchie d'héritage des objets élémentaires de chimie⁴. Un de ces objets est *Atome*, qui décrit les propriétés générales communes à tous les atomes. Le frame *C*, qui représente l'atome de carbone, est une *spécialisation* du frame *Atome* et hérite les propriétés du frame *Atome*. A partir du frame *C*, il est possible de créer un nombre quelconque de carbones particuliers.

⁴Des représentations similaires d'objets chimiques élémentaires ont été présentées dans [AP 90].

3. Le module expert d'ATOMOL

Le module expert est un des quatre modules principaux que comporte le logiciel ATOMOL. Son rôle principal est de gérer la connaissance relative au domaine dans lequel le système travaille. Le module expert recouvre donc une représentation des connaissances et des fonctions d'expertise permettant de manipuler ces connaissances. Les trois objets de base de la représentation sont l'atome, la liaison et la molécule. Ces trois objets suffisent à décrire un modèle constitutionnel de la molécule, qui est considérée comme un graphe non orienté dont les sommets correspondent aux atomes et dont les arêtes correspondent aux liaisons. Les fonctions d'expertise sont des fonctions de perception, qui permettent de dégager des propriétés descriptives et chimiques à partir des caractéristiques structurales des objets chimiques (par exemple la recherche des caractéristiques stéréochimiques ou des fonctionnalités d'une molécule). D'autres fonctions d'expertise permettent de procéder à diverses opérations sur les molécules, comme la visualisation d'une molécule ou la comparaison de deux structures moléculaires. Nous allons préciser dans la suite la structure des objets chimiques élémentaires, atomes, liaisons et molécules.

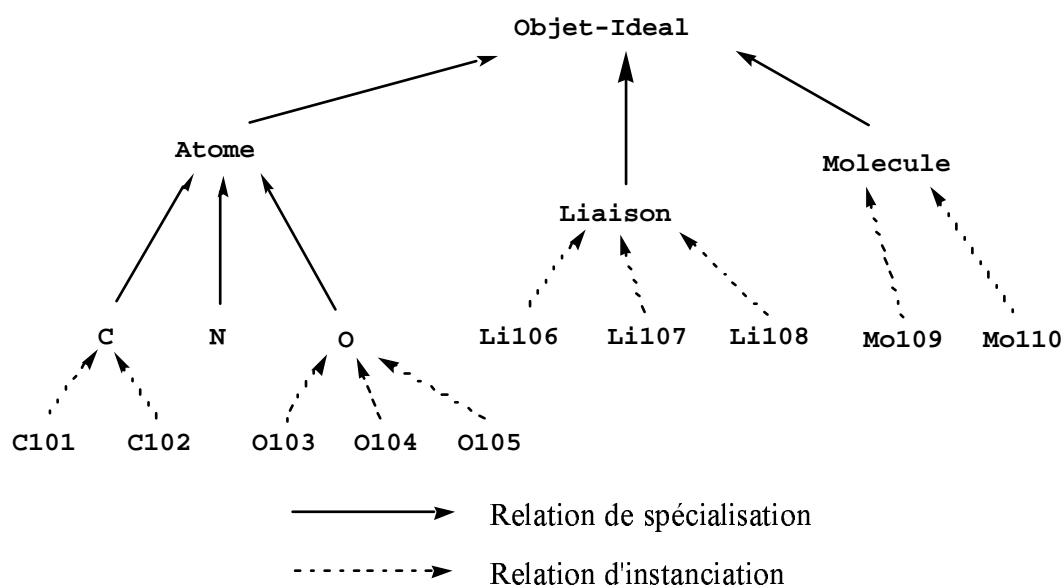


Figure 2. Hiérarchie simplifiée des objets chimiques. Le nom d'une instance est constitué des deux premières lettres de celui du frame, suivies de la valeur d'un compteur qui est incrémenté d'une unité à chaque fois qu'une nouvelle instance est créée. Par exemple, les frames C101 et C102 sont des instances du frame C.

3.1. Description des atomes

Le frame `Atome` définit de manière générique les propriétés communes à tous les atomes (figure 3). Le premier attribut, `numero-atomique`, possède deux facettes de typage qui indiquent que le numéro atomique est un entier appartenant à l'intervalle $[1, 103]$. La masse atomique est un nombre réel compris entre 0 et 260. La charge est une valeur appartenant au domaine $[-3 -2 -1 0 1 2 3]$; par

défaut, l'atome a une charge nulle. La valence⁵ peut prendre une valeur entière allant de 1 à 6. Les vicinales sont les liaisons adjacentes à l'atome. La saturation est un entier qui indique le nombre de liaisons simples que l'atome peut encore former avec d'autres atomes ; elle dépend de la charge de l'atome, de sa valence, et des liaisons adjacentes à l'atome. Par défaut, un atome ne possède aucun voisin ou atome lié. L'attribut `electrons` contient la structure électronique de l'atome, décrite par le nom du gaz rare qui précède l'atome dans le tableau périodique des éléments et par le type des électrons périphériques. La configuration d'un stéréocentre⁶ est un élément du domaine [NIL R S] ; la valeur peut être obtenue par le réflexe `si-besoin calcul-configuration`. Un atome n'a par défaut aucune configuration. L'attribut `molecule` précise le nom de la molécule à laquelle appartient l'atome ; par défaut, il n'appartient à aucune molécule. Enfin, l'attribut `structure-electronique` est le sélecteur d'une méthode appelée `calcul-structure-electronique`, qui donne la structure électronique complète de l'atome. Les notions associées aux attributs `electrons` et `calcul-structure-electronique` sont explicitées sur des exemples au paragraphe 6.

```
(Atome
  (est-un   Objet-Ideal)
  (numero-atomique   ($un entier)
                    ($intervalle [1 103]))
  (masse-atomique    ($un reel)
                    ($intervalle [0 260]))
  (charge   ($domaine [-3 -2 -1 0 1 2 3])
          ($default 0))
  (valence  ($domaine [1 2 3 4 5 6]))
  (vicinales ($des Liaison)
            ($default NIL))
  (saturation ($si-besoin calcul-saturation))
  (voisins   ($des Atome)
            ($si-ajout supprime-saturation)
            ($si-enleve supprime-saturation)
            ($default NIL))
  (électrons ($default NIL))
  (configuration ($si-besoin calcul-configuration)
                ($domaine (NIL R S))
                ($default NIL))
  (molecule ($un Molecule)
            ($default NIL))
  (structure-electronique
    ($methode calcul-structure-electronique)))
```

Figure 3. *Le frame Atome, qui représente le modèle des atomes.*

⁵ La valence d'un atome est le nombre maximal de liaisons auxquelles l'atome peut participer.

⁶ Un stéréocentre est un atome dont les liaisons possèdent une orientation particulière dans l'espace ; selon cette orientation, la configuration du stéréocentre est dite R ou S.

La figure 4 présente une spécialisation du frame `Atome`, le frame `O`, qui représente l'atome d'oxygène. Des facettes de valeur ont été associées aux attributs qui particularisent l'atome d'oxygène.

```
(O
  (est-un Atome)
  (nom ($valeur "oxygène"))
  (numero-atomique ($valeur 8))
  (masse-atomique ($valeur 15.9949))
  (valence ($valeur 2))
  (électrons ($valeur (He 2s 2 2p 4))))
```

Figure 4. *Le frame O qui représente l'atome d'oxygène.*

D'un point de vue pratique, il existe deux possibilités pour calculer la valeur d'un attribut : un réflexe si-besoin qui attache une valeur à l'attribut dès qu'il en a calculé une⁷, une méthode qui calcule systématiquement la valeur. Un réflexe si-besoin est préférable lorsque la fréquence d'accès à la valeur de l'attribut est élevée et que la valeur de l'attribut ne varie pas avec le temps. Une méthode est plutôt utilisée si le calcul de la valeur de l'attribut nécessite des données externes non accessibles directement dans l'objet concerné, si les accès à la valeur de l'attribut sont peu fréquents ou encore si la valeur de l'attribut évolue dans le temps. Ainsi, dans le cas où la valeur de l'attribut est souvent modifiée, une méthode s'impose, car un calcul systématique permet d'obtenir des valeurs constamment à jour. Toutefois, il est toujours possible d'utiliser un réflexe si-besoin, mais dans ce cas, il faut mettre en place des réflexes si-ajout et si-enlève pour maintenir la cohérence des valeurs interdépendantes. C'est le cas pour l'attribut `saturation` : la valeur de la saturation d'un atome est calculée par le réflexe si-besoin `calcul-saturation` (qui n'est activé qu'en cas d'absence de valeur). Si l'environnement d'un atome est modifié par adjonction ou suppression d'atomes voisins, alors la valeur obsolète de l'attribut `saturation` est supprimée par l'intermédiaire des réflexes si-ajout et si-enlève associés à l'attribut `voisins`. La valeur de `saturation` ne variant pratiquement pas dans les activités d'ATOMOL (les molécules sont statiques), un réflexe si-besoin a donc été préféré à une méthode.

3.2. Description des liaisons

Le frame `Liaison` qui décrit les liaisons interatomiques (arêtes du graphe moléculaire) possède quatre attributs (figure 5). L'ordre⁸ de la liaison est un entier

⁷Cette stratégie a été adoptée dans YCHEM, mais la stratégie opposée, à savoir ne pas associer de valeur à l'attribut, est concevable.

⁸L'ordre d'une liaison est en relation avec la force de cette liaison : l'ordre est 1 pour une liaison simple, 2 pour une liaison double et 3 pour une liaison triple. Il s'agit ici d'un ordre formel qui est une simplification de la réalité.

égal à 1, 2 ou 3. Les *extremities* désignent les atomes situés aux extrémités de la liaison. Une liaison ne peut être construite entre deux atomes que si l'ordre de la liaison est inférieur à la saturation de chacun des atomes. C'est le rôle du réflexe *si-possible liaison-possible?* de vérifier cette contrainte : si la condition de saturation n'est pas respectée, les atomes extrémités ne peuvent pas être mis en place et la liaison n'est pas créée. Lorsqu'une liaison est effectivement créée, les attributs *ordre* et *extremities* se voient associer une valeur. Le réflexe *si-ajout +met-a-jour-atomes* attaché à *extremities* entre en action après l'écriture de la valeur de l'attribut : pour chacun des atomes extrémités, il met à jour les valeurs des attributs *voisins* et *vicinales*. Le réflexe *si-enlève -met-a-jour-atomes* agit de manière similaire, au moment de la suppression de la valeur de l'attribut *extremities*. Par défaut, la liaison n'a pas de stéréochimie⁹, mais la valeur de cet attribut peut être calculée par le réflexe *calcul-stereochimie*. Comme pour le frame *Atome*, l'attribut *molecule* indique à quelle molécule appartient la liaison.

```
(Liaison
  (est-un Objet-Ideal)
  (ordre      ($un entier)
             ($domaine [1 2 3]))
  (extremities ($des Atome)
              ($si-possible liaison-possible?)
              ($si-ajout +met-a-jour-atomes)
              ($si-enleve -met-a-jour-atomes))
  (stereochimie ($si-besoin calcul-stereochimie)
                ($domaine [NIL Z E])
                ($default NIL))
  (molecule ($un Molecule)
             ($default NIL)))
```

Figure 5. Le frame *Liaison*.

3.3. Description des molécules

Après les atomes et les liaisons, nous allons décrire le frame *Molecule*, qui représente les graphes moléculaires et qui peut se voir comme un ensemble d'atomes et de liaisons. Le frame *Molecule* possède donc deux attributs qui définissent sa structure : *atomes* et *liaisons* (figure 6). La valeur de ces attributs est constituée de la liste des atomes et des liaisons composant la molécule. La facette *\$si-ajout* associée à l'attribut *atomes* introduit le réflexe *+met-a-jour-nom* dont le but est d'écrire le nom de la molécule dans l'attribut *molecule* de chaque atome appartenant à la molécule. Si on supprime des atomes dans la molécule, le réflexe *-met-a-jour-nom* supprime la valeur de l'attribut *molecule* pour chacun d'entre eux. Les mêmes réflexes sont utilisés lors de l'adjonction ou la suppression de liaisons. Les

⁹Une liaison possède une stéréochimie si les positions des groupements d'atomes liés à ses extrémités confèrent une géométrie particulière à la molécule. Selon la nature et la position des groupements, la stéréochimie d'une liaison peut être Z ou E.

attributs `cycles`, `cycles-aromatiques` et `classes-de-symetrie` mémorisent les caractéristiques chimiques de la molécule. Leurs valeurs sont calculées par des réflexes si-besoin qui permettent de procéder à la *perception chimique* de la molécule. Les fonctions de perception chimique permettent l'analyse des molécules et entrent en jeu notamment pour comparer les molécules proposées par les étudiants dans les exercices à des molécules de référence. Les informations obtenues lors de la perception sont également nécessaires pour visualiser les molécules dans les énoncés des exercices, et ce dans les divers modes de représentation des structures moléculaires.

```
(Molecule
  (est-un Objet-Ideal)
  (atomes ($des Atome)
    ($default NIL)
    ($si-ajout +met-a-jour-nom)
    ($si-enleve -met-a-jour-nom))
  (liaisons ($des Liaison)
    ($default NIL)
    ($si-ajout +met-a-jour-nom)
    ($si-enleve -met-a-jour-nom))
  (cycles ($si-besoin perception-cycles))
  (cycles-aromatiques
    ($si-besoin perception-aromaticite))
  (classes-de-symetrie
    ($si-besoin perception-symetrie)))
```

Figure 6. *Le frame Molecule.*

Afin d'expliciter les relations qui existent entre atomes, liaisons et molécules, nous allons décrire la création d'une instance de `Molecule`. Cette création se déroule en trois temps : création des atomes, puis des liaisons, puis de la molécule elle-même. Voyons le cas d'un exemple simple : une molécule d'eau (figure 7). La première étape est la création de deux instances du frame `H`, `H1` et `H2` et d'une instance du frame `O`, `O3`. Ensuite, une liaison `Li4` d'ordre 1 dont les extrémités sont les atomes `H1` et `O3` est créée. Cette création s'accompagne de la modification de la valeur des attributs `voisins` et `vicinales` des atomes `H1` et `O3`. `H1` a `O3` pour voisin et `Li4` pour vicinale. Le déroulement est similaire pour la création de la deuxième liaison simple `Li5` entre `H2` et `O3`. Enfin, une instance du frame `Molecule` est créée, `Mo6`, à laquelle sont attribuées la liste d'atomes (`H1 H2 O3`) et la liste de liaisons (`Li4 Li5`). La mise en place des atomes et des liaisons est suivie de l'écriture du nom de la molécule, `Mo6`, dans l'attribut `molecule` de chacun des constituants.

4. Le module d'enseignement

Le module d'enseignement contient la description des tâches qui sont proposées à l'étudiant et qui sont soit des exercices, soit des problèmes. Le module d'enseignement s'articule autour de deux niveaux : un niveau de tâches génériques qui

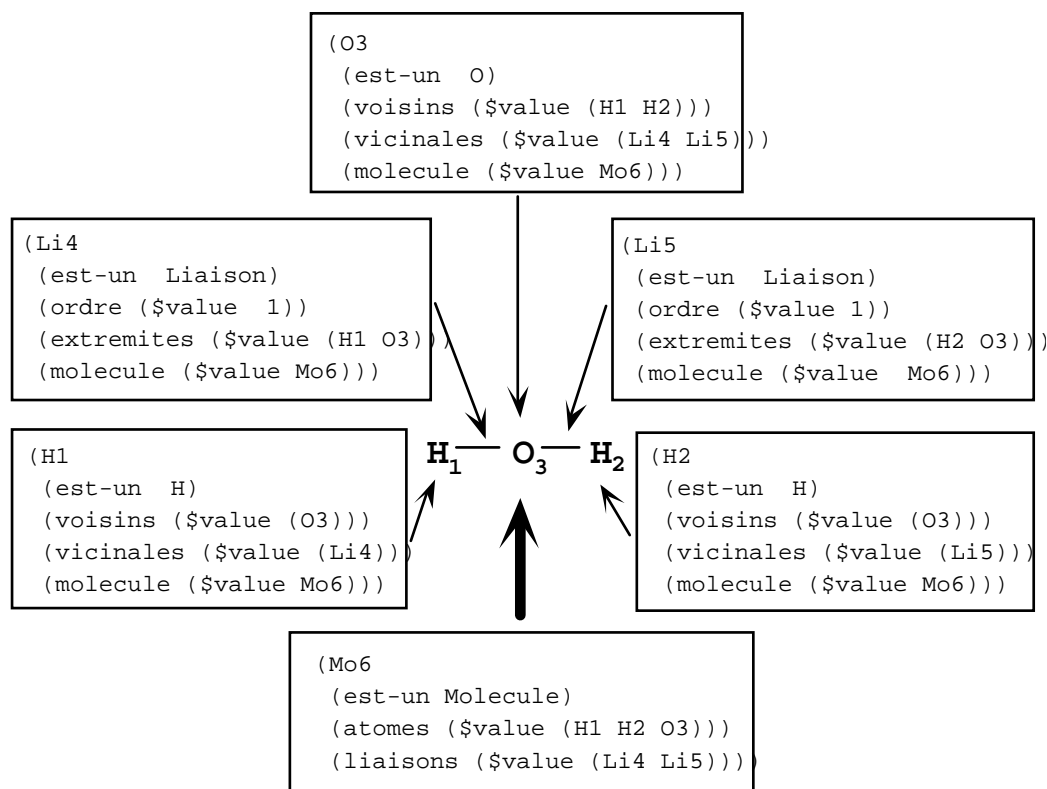


Figure 7. Création d'une molécule d'eau.

peuvent s'appliquer à un domaine quelconque, et un niveau de tâches spécifiques dédiées à la chimie. Dans une première partie, nous définissons la notion de problème telle qu'elle a été envisagée pour ATOMOL. Ensuite, nous décrivons la structure du module d'enseignement et de ses principales composantes : tâche, exercice et problème. Durant cette description, nous donnons quelques exemples d'exercices et de problèmes de chimie.

4.1. La notion de problème

D'une manière générale, un problème est en relation avec l'apprentissage d'un concept de chimie organique. Un tel concept possède un degré de complexité lié à l'ensemble de connaissances qu'il est nécessaire d'avoir pour pouvoir appréhender le concept. Ainsi, moins un concept nécessite de connaissances, moins il est *complexe* (et plus il est élémentaire), et réciproquement, plus un concept nécessite de connaissances, plus il est difficile à apprendre, et plus il s'appuie sur la connaissance de concepts élémentaires. L'apprentissage d'un concept se traduit donc par un ensemble de tâches de base qui sont en rapport avec l'apprentissage de concepts élémentaires.

Pratiquement, un problème se définit par rapport à un concept à apprendre par l'étudiant comme une composition d'exercices, qui correspondent chacun à une tâche de base, donc à un concept élémentaire à apprendre. Dans la composition d'un

problème, certaines séquences d'exercices peuvent être isolées parce qu'elles forment un ensemble cohérent permettant d'appréhender un concept de degré de complexité inférieur à celui du concept auquel le problème s'intéresse dans sa globalité. Le concept de degré de complexité inférieur s'appelle un sous-concept et la séquence d'exercices un sous-problème. C'est l'enseignant qui décide du degré de complexité d'un concept, et en particulier d'isoler des séquences d'exercices en problèmes ou sous-problèmes.

Un problème est représenté par un graphe orienté sans circuits appelé *graphe des tâches*. Les sommets du graphe symbolisent des tâches élémentaires et les arcs du graphe symbolisent la relation "*est un prérequis de*", qui permet de classer les tâches par leur degré de complexité. Plus précisément, la tâche symbolisée par un sommet S1 est un prérequis de la tâche symbolisée par un sommet S2 si le concept associé à S1 doit être maîtrisé pour que le concept associé à S2 puisse être lui-même maîtrisé (le concept associé à S1 est moins complexe que le concept associé à S2). La relation "*est un prérequis de*" est antisymétrique et transitive, mais non réflexive et non symétrique (pas de boucle ni de circuit dans le graphe des tâches). La notion de prérequis existe dans de nombreux systèmes d'enseignement assisté par ordinateur, soit pour exécuter un retour-arrière quand une ou plusieurs erreurs ont eu lieu pendant la résolution d'une tâche, soit comme filtre d'accès à une tâche [REG 88] [COR+ 88][GAV 91].

Un exemple de graphe de tâches est donné à la figure 8 : le problème Pb1 est constitué de 4 exercices et d'un sous-problème Pb2, le sous-problème Pb2 étant lui-même constitué de 3 exercices.

Le graphe des tâches représente en réalité l'espace d'états associé au problème à résoudre par l'étudiant [NIL 80]. En s'appuyant sur ce parallèle, il est possible d'envisager plusieurs modes de résolution de problèmes correspondant à autant de stratégies possibles de parcours de l'espace d'états. Ainsi, un premier mode de résolution consiste à guider l'étudiant vers le but d'un problème en commençant par les tâches les plus élémentaires et en continuant par des tâches de plus en plus complexes faisant appel à des résultats antérieurs (démarche *synthétique*). Ce premier mode de résolution est dit *dirigé par les données*. Pour le problème Pb1 de la figure 8, un parcours possible est : Ex5 - Ex4 - Ex3 - Ex2 - Ex8 - Ex7 - Ex6 - Ex1. Le deuxième mode de résolution consiste à partir du but d'un problème pour renvoyer l'étudiant à des tâches de complexité inférieure en cas d'échec (démarche *analytique*). Ce second mode de résolution est dit *dirigé par le but avec (éventuel) retour aux prérequis*. Ainsi, si l'exercice Ex1 est proposé à l'étudiant et qu'il se trompe, il lui est demandé de résoudre les exercices Ex6 et Ex2, et les exercices de complexité inférieure en cas de nouvel échec. Un troisième mode de résolution consiste à reconstruire le graphe des tâches associé au problème. Les exercices sont proposés en vrac à l'étudiant qui doit les "ordonner", autrement dit, suivre une démarche qui soit en accord avec la relation "*est un prérequis de*". En particulier, si l'étudiant choisit un exercice d'un certain degré de complexité alors que les exercices de complexité inférieure n'ont pas été traités, l'étudiant commet une erreur.

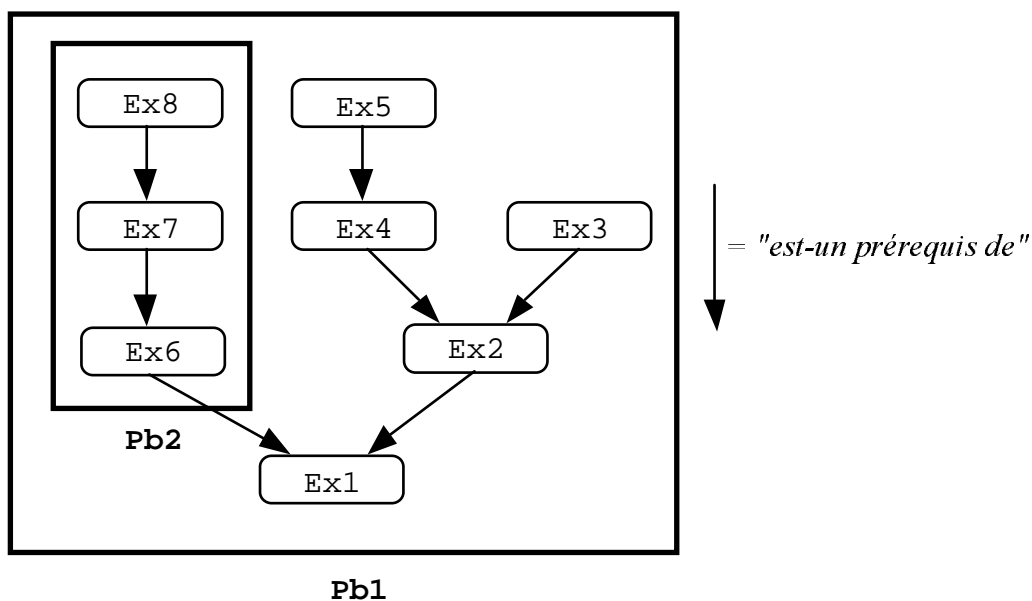


Figure 8. Exemple de représentation d'un problème.

En résumé, la représentation d'un problème qui a été choisie, à savoir le graphe des tâches, s'adapte bien à l'application de méthodes de résolution multi-étapes (comportant une succession d'étapes de résolution élémentaires). Les différentes méthodes de résolution correspondent à différents modes de parcours possibles du graphe des tâches. L'étudiant se voit ainsi proposé plusieurs points de vue sur un même problème. Le mode "dirigé par les données" est plutôt adapté à une phase d'apprentissage, où l'étudiant aborde le problème pour la première fois. La reconstruction d'un problème nécessite pour sa part une bonne connaissance de la structure du problème et des relations existant entre les divers composants du problème. Elle peut donc être utilisée pour contrôler les acquis d'un étudiant. Le mode "dirigé par le but avec retour aux prérequis" s'impose lorsque l'étudiant s'intéresse à un point particulier dans un problème (matérialisé par un ou plusieurs exercices). Si le renvoi systématique à des prérequis peut apporter une aide dans de nombreux cas, il n'est toutefois pas certain que l'erreur soit due à une méconnaissance des prérequis. La source de l'erreur peut résider dans une toute autre cause : méconnaissance des règles permettant de résoudre l'exercice, mauvaise compréhension de l'énoncé, etc. Nous ne disposons actuellement d'aucun moyen permettant de déterminer avec précision si la cause de l'erreur est due ou n'est pas due à une méconnaissance des prérequis.

Soulignons toutefois que le parcours du graphe des tâches ne correspond pas véritablement à une résolution de problème. En effet, ATOMOL ne peut pas rechercher de solutions particulières autres que celles qui sont déjà implicitement codées dans le graphe des tâches. Il n'est donc pas possible de proposer d'autres modes de résolution que ceux décrits ci-dessus. En cela, le fonctionnement d'ATOMOL est limité, car les caractéristiques d'un système de résolution de problèmes, comme en dispose le programme GEORGE par exemple [CCC 86], lui font défaut.

4.2. La structure du module d'enseignement : tâches, exercices et problèmes

Le module d'enseignement est décrit par une hiérarchie d'objets, dans laquelle on distingue une partie générique, composée des frames Tache, Probleme, Exercice, Repondre-Au-Clavier, Completer, Choisir, Dessiner, et une partie spécifique au domaine de la chimie (figure 9). La racine de cette hiérarchie est le frame Tache, qui contient les caractéristiques communes à toutes les tâches. Deux spécialisations du frame Tache sont les frames Probleme et Exercice qui décrivent la constitution et le déroulement d'un problème ou d'un exercice, en général. Les exercices génériques Repondre-Au-Clavier, Completer, Choisir, Dessiner décrivent chacun un déroulement particulier d'exercice. Les exercices et les problèmes de chimie sont des spécialisations des exercices et du problème génériques.

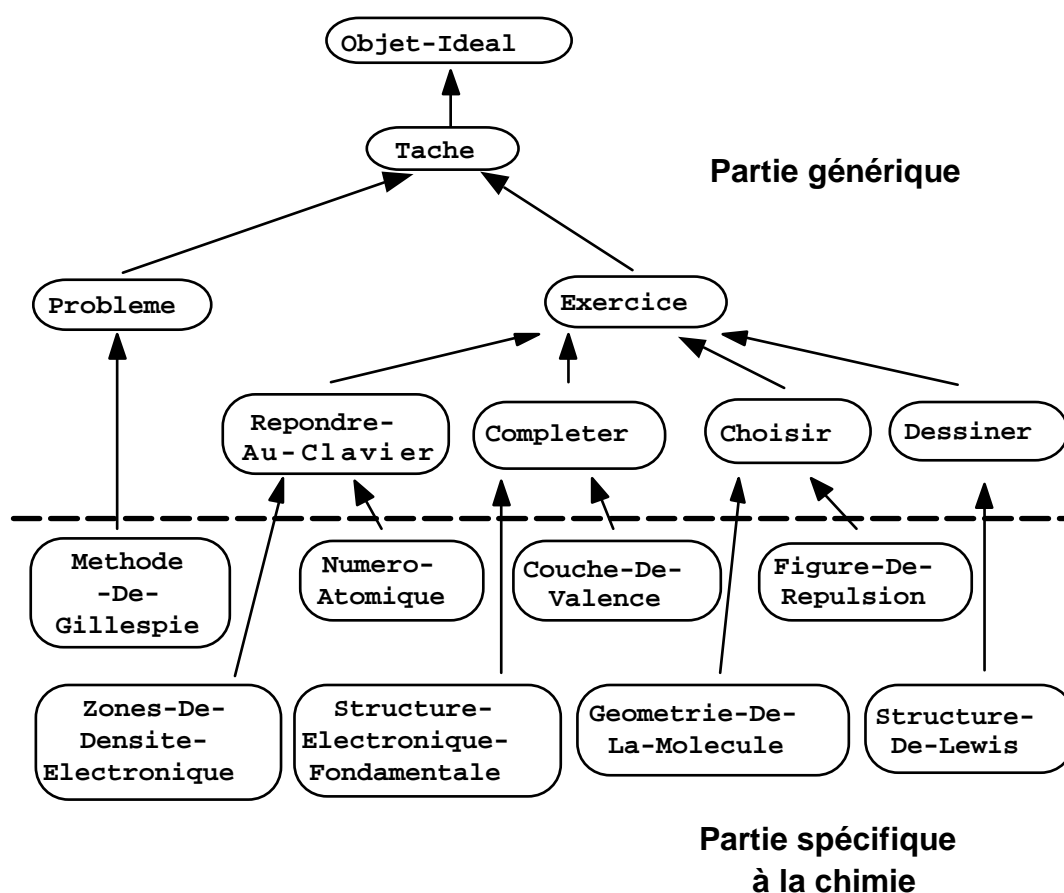


Figure 9. Le module d'enseignement.

Nous présentons maintenant certains éléments génériques du module d'enseignement : le frame Tache, le frame Exercice et une de ses spécialisations, l'exercice Choisir, et nous terminerons par la description du frame Probleme. Parallèlement, nous donnons des exemples d'application de ces éléments génériques à la chimie.

4.2.1. Les tâches

Le frame `Tache` spécifie les propriétés communes à toutes les tâches (figure 10). Pour chaque tâche, trois niveaux de difficulté sont prévus, le premier niveau étant attribué à une tâche élémentaire, le deuxième à une tâche de difficulté moyenne, et le troisième aux tâches les plus ardues. Ces trois niveaux sont matérialisés par trois listes de données appelées `donnees1`, `donnees2` et `donnees3`. Par exemple, supposons qu'une tâche consiste à demander le numéro atomique d'un atome (exercice appelé `Numero-Atomique` dans la suite), les listes de données correspondant aux trois niveaux de difficultés sont les suivantes :

- le niveau 1 pour les atomes des deux premières périodes du tableau périodique dont les numéros atomiques sont habituellement le mieux mémorisé :

```
donnees1 = (H He Li Be B...)
```

- le niveau 2 pour les atomes de la troisième période et quelques autres atomes qui sont rencontrés fréquemment :

```
donnees2 = (Na Ar Br...)
```

- le niveau 3 pour les autres atomes (atomes pour lesquels même les spécialistes consultent le tableau périodique) :

```
donnees3 = (Sc Ti V...)
```

Les atomes sont donc traités par classes de difficulté et non individuellement. Plutôt que de créer une seule tâche, `Numero-Atomique`, avec trois listes de données, il aurait été possible de créer une tâche pour chacun des niveaux de difficulté, ou même une tâche pour chaque atome. Nous avons procédé de cette manière puisque le déroulement de l'exercice est le même quel que soit l'atome en question.

Le niveau de difficulté d'un problème est le maximum des niveaux de difficulté des exercices ou des sous-problèmes qui le composent.

Lors de l'instanciation d'une tâche proposée à un étudiant, le système `ATOMOL` affecte à chacun des attributs `niveau`, `donnees` et `appelant`, une valeur particulière, qui est fixée en fonction de l'historique de l'étudiant (voir § 5). Le niveau de difficulté d'une tâche est 1, 2 ou 3. L'attribut `donnees` contient les données particulières d'une instance d'exercice ou de problème. Par exemple, pour l'exercice concernant le numéro atomique du sodium `Na`, qui est une instance de l'exercice `Numero-Atomique`, la valeur de l'attribut `niveau` est 2 et celle de l'attribut `donnees` est `Na`. Les données sont contenues dans des fichiers et ne sont chargées que par nécessité, après l'activation d'un réflexe si-besoin associé à chacun des attributs qui lit le fichier de données, et ceci afin de ne pas surcharger la mémoire centrale de l'ordinateur avec les données de l'ensemble des tâches. L'attribut `erreurs` est un entier égal au nombre d'erreurs commises par l'étudiant lors de la résolution d'une tâche. Enfin, si l'exécution de la tâche a été provoquée par une autre tâche (par exemple lorsqu'un exercice est appelé par un problème), le nom de cette dernière est écrit dans l'attribut `appelant`. Un dernier attribut concernant les données est la méthode `savegarder-donnees` qui sert à sauvegarder les listes de données quand elles ont été modifiées par l'enseignant, ce dernier disposant d'outils permettant d'ajouter des données aux exercices et aux problèmes, ou d'en supprimer.


```

(Tache
  (est-un Objet-Ideal)
  (donnees1 ($si-besoin lire-donnees1))
  (donnees2 ($si-besoin lire-donnees2))
  (donnees3 ($si-besoin lire-donnees3))
  (niveau ($domaine [1 2 3]))
  (donnees ($default NIL))
  (appelant ($un Tache)
    ($default NIL))
  (erreurs ($un entier)
    ($default 0))
  (sauvegarder-donnees
    ($methode sauvegarder-donnees)))

```

Figure 10. *Le frame Tache.*

4.2.2. *Les exercices*

Le frame `Exercice` contient la description générale d'un exercice et possède quatre types d'attributs (figure 11). Les premiers, `type-fenetre`, `zone-enonce`, `zone-commentaires` et `icomes` servent à spécifier l'environnement graphique d'un exercice. Les seconds, `creation`, `destruction`, `resolution`, `presentation`, `terminaison` et `notation` sont en rapport avec les opérations exécutées lors du déroulement d'un exercice depuis la création d'une instance d'exercice jusqu'à sa destruction, en passant par la résolution de l'exercice proprement dite. L'attribut `prerequis` sert à définir les liens générés par la relation *est-un-prerequis-de*. Les derniers attributs, `demandes-aide` et `demande-solution`, servent à mémoriser les actions d'un étudiant qui essaie de résoudre l'exercice.

L'environnement graphique général associé à un exercice est défini par les quatre premiers attributs. L'attribut `type-fenetre` spécifie l'aspect de la fenêtre présentant un exercice. Deux autres attributs servent à délimiter les zones dans la fenêtre : l'attribut `zone-enonce` où est affiché l'énoncé et l'attribut `zone-commentaires` où sont annoncés les messages d'erreur et les aides. Un dernier attribut, `icomes`, contient le nom des icônes associées à l'exercice, et qui sont par défaut les trois icônes `Icone-Fini`, `Icone-Aide`, et `Icone-Sol` (figure 12). L'étudiant doit sélectionner l'icône *FINI* pour valider sa réponse. La sélection de l'icône *AIDE* entraîne l'affichage d'une ou plusieurs aides. L'étudiant peut sélectionner l'icône *SOL* s'il ne parvient pas à résoudre l'exercice, et la solution est alors donnée par le logiciel. Comme cet environnement par défaut est hérité par l'ensemble des exercices, la cohérence dans leur présentation est garantie : les différentes zones ont la même disposition et la fonction des icônes est la même dans tous les exercices. Ceci permet à l'étudiant de s'adapter rapidement à l'interface du logiciel. Cette façon de faire s'apparente à ce qui se pratique couramment dans le monde des micro-ordinateurs où

```

(Exercice
  (est-un Tache)
  (type-fenetre ($default Fenetre-exercice))
  (zone-enonce ($default (0 50 30 500)))
  (zone-commentaires ($default (270 50 300 500)))
  (icônes ($default (Icône-Aide Icône-Fini Icône-Sol)))

  (creation ($methode creation-exercice))
  (destruction ($methode destruction-exercice))
  (resolution ($methode resolution-exercice))
  (presentation ($methode presentation-exercice))
  (terminaison ($methode fin-exercice))
  (notation ($methode notation-exercice))

  (prerequis ($des Exercice)
             ($default NIL))

  (demandes-aide ($un entier)
                 ($default 0))
  (demande-solution ($domaine [NIL t])
                    ($default NIL))

```

Figure 11. *Le frame Exercice.*

FINI

AIDE

SOL

Quel est le numéro atomique de l'atome de carbone (C) ? Entrez votre proposition et validez en sélectionnant l'icône FINI.

Votre proposition : 6

Figure 12. *L'exercice concernant le numéro atomique.*

la grande majorité des logiciels se présentent sous le même aspect graphique (éditeur de texte, éditeur graphique, etc.)

Le déroulement d'un exercice est également commun à tous les exercices, afin de favoriser une bonne et rapide adaptation de l'apprenant. En effet, toutes les méthodes qui assurent le déroulement d'un exercice sont définies au niveau de l'exercice générique : sa création, sa présentation, sa résolution, sa terminaison, sa notation et sa destruction.

L'attribut `prerequis` d'un exercice contient la liste des noms des exercices qui constituent ses prérequis. Par exemple, la valeur de cet attribut pour l'exercice `Ex1` de la figure 8 est (`Ex6 Ex2`).

Les deux derniers attributs du frame `Exercice` servent à l'évaluation du travail de l'étudiant. L'attribut `demandes-aide`, qui vaut zéro par défaut, est augmenté d'une unité chaque fois que l'étudiant demande de l'aide. La valeur de l'attribut `demande-solution`, qui vaut `NIL` par défaut, indique `vrai` si l'étudiant a demandé la solution. Les valeurs de ces attributs, ainsi que celle de l'attribut `erreurs` hérité du frame `Tache` sont utilisées pour attribuer une note à l'étudiant en fin d'exercice ; elles sont également inscrites dans l'historique du travail de l'étudiant (cf. § 5).

Le frame `Exercice` décrit la présentation et le déroulement d'un exercice en général. Les spécialisations de `Exercice` modélisent des déroulements particuliers d'exercices : répondre à l'aide du clavier, choisir parmi plusieurs propositions, compléter, dessiner, etc. Examinons en détail le frame `Choisir`, puis une de ces spécialisations en chimie, l'exercice `Geometrie-De-La-Molecule`.

L'exercice générique Choisir

Deux méthodes supplémentaires sont toujours associées à une spécialisation de `Exercice` : la méthode `action` et la méthode `controle-solution`. La méthode `action` interprète les opérations effectuées par l'étudiant : la sélection des icônes, les caractères entrés au clavier, les mouvements de la souris. Dans le cas de l'exercice décrit par le frame `Choisir`, l'étudiant doit sélectionner des boutons¹⁰ parmi ceux qui sont affichés à l'écran, en utilisant la souris (figure 14).

```
(Choisir
  (est-un Exercice
    (boutons ($default NIL))
    (presentation ($methode presentation-choix))
    (action ($methode action-choix))
    (controle-solution ($methode controle-choix))
    (boutons-choisis ($default NIL)))
```

La méthode `controle-solution` vérifie que le travail de l'étudiant correspond à la réponse attendue et donne un diagnostic. Le contrôle de la solution est fait par une procédure qui analyse la réponse donnée par l'étudiant. Par défaut, la méthode `controle-solution` compare le contenu du bouton choisi par l'étudiant à la réponse correcte et indique simplement si le choix de l'étudiant est juste ou faux. Une autre possibilité serait de diagnostiquer une éventuelle erreur par comparaison à des classes d'erreur contenues dans un catalogue, comme dans le système `DEBUGGY` qui traite de la soustraction [BUR 82] ou dans le système `FRACT` pour les fractions [DDG 89].

L'environnement et la méthode de présentation d'un exercice sont redéfinis, si nécessaire. C'est le cas de l'exercice `Choisir` : l'attribut `boutons` contient la liste des boutons et la méthode de présentation est particularisée par rapport à celle du frame `Exercice`, puisque les boutons doivent être affichés, en plus des icônes et de l'énoncé de l'exercice (figure 14).

¹⁰Un bouton est défini par une zone rectangulaire et son contenu.

Chaque exercice générique possède également un attribut qui sert à enregistrer les actions qui ont été effectuées par l'étudiant, afin de pouvoir interrompre momentanément le déroulement de l'exercice, puis le reprendre ultérieurement dans le même état, par exemple s'il a été demandé à l'étudiant de résoudre un prérequis de l'exercice. Dans le cas de l'exercice Choisir, ce rôle est dévolu à l'attribut boutons-choisis qui contient la liste des boutons sélectionnés par l'étudiant.

Un exercice spécifique : Geometrie-De-La-Molecule

Considérons par exemple l'exercice spécifique Geometrie-De-La-Molecule qui est une spécialisation de l'exercice générique Choisir (figure 13). Les différentes géométries possibles sont proposées dans des boutons ; l'étudiant choisit la géométrie qui convient en sélectionnant le bouton correspondant à l'aide de la souris (figure 14).

Chaque donnée de l'exercice comprend le nom d'une molécule et sa géométrie. Les différents niveaux de l'exercice sont fonction du type des atomes présents dans la molécule traitée (cf. § 6). L'environnement graphique est particularisé par rapport à celui hérité de Exercice par les attributs titre-fenetre, enonce et boutons. Dans un exercice spécifique sont également définis les aides disponibles ainsi que le texte-final à afficher lorsque l'exercice est achevé.

```
(Geometrie-De-La-Molecule
 (est-un Choisir)
 (donnees1 ($valeur ((BF3 Triangle)...)))
 (donnees2 ($valeur ((PF3 Pyramide trigonale)...)))
 (donnees3 ($valeur ((SF6 Octaèdre)...)))

 (titre-fenetre ($valeur "Géométrie de la molécule"))
 (icomes ($valeur (Icône-Aide Icône-Sol)))
 (enonce ($valeur (Quelle est la géométrie de la
                   molécule (donnee 1) ?)))
 (boutons ($valeur ((Linéaire 50 50 70 150)
                   (Forme T .....)))
 (aides ($valeur ((La géométrie d'une molécule
                   s'obtient à partir de la figure
                   de répulsion en ne prenant en
                   compte que les doublets))))
 (texte-final ($valeur (La géométrie de la molécule
                       (donnee 1) est : (donnee 2).
                       (dessiner-molecule (donnee 1))))))
```

Figure 13. *Le frame Geometrie-De-La-Molecule.*

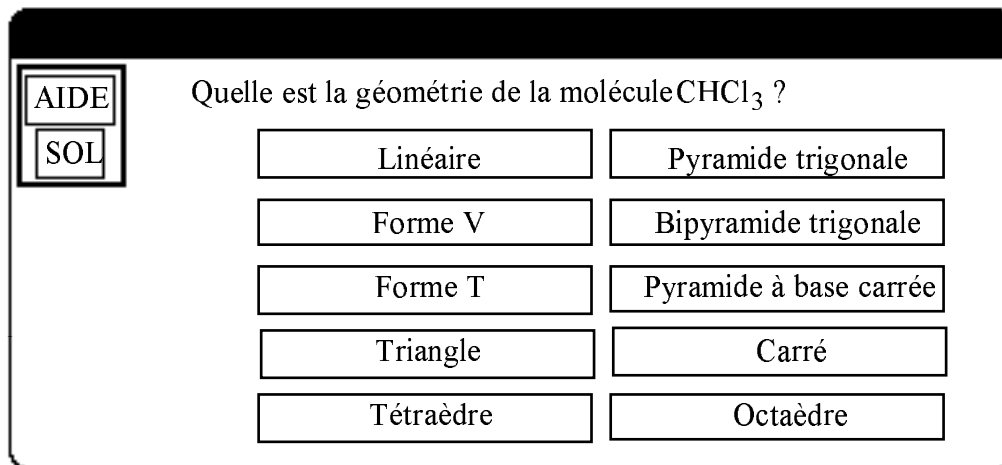


Figure 14. Choix de la géométrie de la molécule CHCl_3 .

4.2.3. Les problèmes

Un problème, représenté par l'objet générique `Probleme`, est un objet composite. Les composantes sont des sous-tâches, exercices ou sous-problèmes, dont les noms sont mémorisés par l'attribut `sous-taches`.

```
(Probleme
  (est-un Tache)
  (sous-taches ($des Tache)
               ($default NIL))
  ...
  ... )
```

Du point de vue du code, un problème est représenté par un objet du même type que `Exercice` (figure 11) : un environnement graphique par défaut est défini dans le frame `Probleme`, qui comprend une fenêtre où est affiché l'énoncé du problème et un document qui contient tous les résultats auxquels l'étudiant a abouti lors de la résolution des différentes sous-tâches. Les autres attributs sont les sélecteurs de méthodes qui sont activées lors du déroulement d'un problème.

Comme pour les exercices, la cohérence dans la présentation et le déroulement des problèmes est assurée, puisque tous les problèmes spécifiques sont des spécialisations de `Probleme` dont ils héritent l'environnement graphique et les méthodes nécessaires à leur déroulement.

Un problème spécifique : Methode-De-Gillespie

Ainsi, le problème `Methode-De-Gillespie` est un exemple de problème spécifique dont l'objet est de déterminer la structure électronique puis la géométrie d'une molécule. Il est composé de sept exercices dont la liste des noms constitue la valeur de l'attribut `sous-taches` (figure 15). Comme toutes les tâches, ce problème

```

(Methode-De-Gillespie
  (est-un Probleme)
  (sous-taches
    ($valeur (Numero-Atomique
              Structure-Electronique-Fondamentale
              Couche-De-Valence
              Structure-De-Lewis
              Zones-De-Densite-Electronique
              Figure-De-Repulsion
              Geometrie-De-La-Molecule)))
  (donnees1 ($valeur ((BF3 B)
                      (NH3 N)...)))
  (donnees2 ($valeur ((PF3 P)...)))
  (donnees3 ($valeur ((IF5 I)...)))
  (titre-fenetre ($valeur "Méthode de Gillespie"))
  (enonce ($valeur (Le but du problème est de
                    déterminer la répartition des
                    électrons de valence et la
                    géométrie de la molécule (donnee 1).
                    On précise que (donnee 2) est
                    l'atome central.))))

```

Figure 15. *Le problème Methode-De-Gillespie.*

comporte trois listes de données matérialisant les niveaux de difficulté. Les deux derniers attributs définissent le titre de la fenêtre associée au problème et l'énoncé du problème. Le déroulement du problème *Methode-De-Gillespie* est détaillé au paragraphe 6.

5. L'historique de l'étudiant

L'historique de l'étudiant sert à déterminer les acquis de l'étudiant et les objectifs qu'il doit encore atteindre. Il est utilisé par le module d'enseignement pour la sélection des tâches devant être proposées à l'étudiant. Il est consulté quand l'étudiant débute une nouvelle séance de travail, et il est mis à jour pendant la séance de travail, chaque fois qu'une tâche est achevée ou que l'étudiant a commis une erreur. L'historique d'un étudiant est conservé dans des fichiers de données. Un tel historique aurait pu être représenté par un frame s'il avait été possible de gérer des objets persistants dans le cadre d'ATOMOL, ce qui n'est pas le cas actuellement.

5.1. Composition de l'historique

L'historique de chaque étudiant qui utilise ATOMOL est sauvegardé dans deux fichiers où les informations qui le concernent sont enregistrées dans l'ordre chronologique. Le premier fichier est un *historique des tâches* réalisées par l'étudiant et des principaux événements qui ont eu lieu lors du travail sur le logiciel (début de séance, changement de mode de travail). Quand l'étudiant a terminé une tâche,

certaines informations sauvegardées dans l'historique servent à une identification ultérieure de cette tâche (nom de la tâche et données utilisées), d'autres décrivent la façon dont s'est déroulée la tâche (note, nombre d'erreurs commises, nombre d'aides demandées, demande éventuelle de la solution). Le deuxième fichier est l'*historique des erreurs* qui ont été commises par l'étudiant. Ce fichier est mis à jour chaque fois que l'étudiant fait une erreur, par une description exacte de la configuration de l'exercice au moment où l'erreur a été commise. Ce fichier est destiné à l'enseignant qui peut consulter les réponses données par les étudiants quand des erreurs ont été détectées. Le contenu de l'historique des erreurs peut être supprimé après avoir été consulté, alors que celui de l'historique des tâches est permanent.

En fait, l'historique est un des éléments classiques du modèle de l'apprenant, qui contient aussi des informations décrivant le niveau de connaissances de l'apprenant et les traits particuliers de l'apprenant [DED 86][SEL 87][NV 88]. Le rôle essentiel du modèle de l'apprenant est l'individualisation du dialogue, notamment en ce qui concerne le choix des tâches à proposer à l'apprenant et la correction des connaissances erronées. Le plus souvent, l'historique est une contraction des sessions antérieures et est utilisé pour évaluer les performances de l'apprenant. L'historique d'ATOMOL se veut cependant "complet" : il conserve, dans l'ordre chronologique, toutes les informations relatives au travail de l'apprenant.

5.2. L'historique et l'apprenant

Lorsqu'un étudiant débute une séance de travail avec le logiciel ATOMOL, son historique des tâches est consulté afin de créer une "image" des travaux qu'il a effectués. Cette "image" est utilisée par le module d'enseignement lors du choix des tâches à proposer à l'étudiant et est construite de manière à suivre trois règles élémentaires : tenir compte des notions déjà acquises par l'étudiant, insister sur celles qui ne sont pas encore acquises et éviter de proposer plusieurs fois les mêmes tâches. L'historique est alors parcouru afin d'établir une priorité dans les travaux qui seront proposés à l'étudiant.

Chacune des listes `donnees1`, `donnees2` et `donnees3` de chaque tâche est décomposée en quatre sous-ensembles :

- $E = \{\text{données correspondant à des tâches où l'étudiant a commis une ou plusieurs erreurs}\}$,
- $A = \{\text{données correspondant à des tâches où l'étudiant a demandé de l'aide puis a résolu la tâche sans commettre d'erreur}\}$,
- $J = \{\text{données correspondant à des tâches qui n'ont jamais été traitées par l'étudiant ou des tâches pour lesquelles l'étudiant a demandé la solution}\}$,
- $P = \{\text{données correspondant à des tâches déjà parfaitement résolues par l'étudiant}\}$.

Lors du choix d'une tâche à proposer à l'étudiant, le module d'enseignement sélectionne en priorité une donnée dans l'ensemble E . Si E est vide, il choisit une donnée dans A , et à défaut dans J , puis dans P .

Pour l'exercice *Numero-Atomique*, par exemple, la liste des données de niveau 1 est : $donnees1 = (H\ He\ Li\ Be\ B\ C\ N\ O\ F\ Ne)$. Supposons qu'un étudiant utilise ATOMOL pour la première fois, les quatre sous-ensembles de $donnees1$ sont $E = \emptyset$, $A = \emptyset$, $J = \{H\ He\ Li\ Be\ B\ C\ N\ O\ F\ Ne\}$ et $P = \emptyset$; E , A , et P sont vides et le module d'enseignement choisit alors aléatoirement des données dans J . Supposons également que lors de cette première séance l'étudiant résolve parfaitement les exercices concernant les numéros atomiques de H, C et O, commette une erreur pour B, demande de l'aide pour N et Ne, et n'aborde pas les autres. Quand l'étudiant débute une deuxième séance avec ATOMOL, les sous-ensembles sont $E = \{B\}$, $A = \{N\ Ne\}$, $J = \{He\ Li\ Be\ F\}$ et $P = \{H\ C\ O\}$: le module d'enseignement propose alors les exercices traitant du numéro atomique de B, puis de N, puis de Ne, puis de He, etc.

Ces règles de priorité sont utilisées dans les deux modes de travail possibles : *travail imposé* ou *travail libre*. En mode de travail imposé, la succession des tâches proposées à l'étudiant suit une progression qui a été définie au préalable par l'enseignant à l'aide d'un utilitaire intégré au logiciel. Une partie de la progression peut par exemple être celle présentée dans le tableau ci-dessous.

N°	Type	Tâche	Niveau	Nombre
1	Exercice	Structure électronique fondamentale	1	10
2	Exercice	Structure électronique fondamentale	2	8
3	Exercice	Structure électronique fondamentale	3	6
4	Retour aux prérequis	Couche de valence	1	10
5	Retour aux prérequis	Couche de valence	2	8
6	Retour aux prérequis	Couche de valence	3	6
7	Problème dirigé	Structure et géométrie de la molécule	1	5
8	Problème reconstruit	Structure et géométrie de la molécule	2	5
9

La progression peut contenir des exercices simples, des exercices à résoudre en mode retour aux prérequis et des problèmes en mode dirigé ou reconstruction. Les trois premiers items de la progression indiquent que l'étudiant doit résoudre 10 exercices concernant la structure électronique fondamentale au niveau 1, puis 8 au niveau 2 et 6 au niveau 3. Ces trois étapes étant franchies, le module d'enseignement propose à l'étudiant l'exercice qui traite de la couche de valence, en mode retour aux prérequis, pour les trois niveaux de difficulté. Ensuite, l'étudiant peut aborder le problème intitulé "Structure et géométrie de la molécule", d'abord en mode dirigé, puis en mode reconstruit.

Si l'étudiant ne veut pas suivre la progression imposée par le module pédagogique, il peut choisir le mode de travail libre et sélectionner la tâche qui l'intéresse. A l'intérieur de ce mode de travail, deux options différentes sont possibles concernant le choix des données pour une tâche particulière. Pour la première option, le choix des données est contrôlé par le module d'enseignement, conformément aux règles de priorité énoncées ci-dessus. Avec la deuxième option, l'étudiant a la possibilité de

choisir lui-même des données particulières pour la tâche (pour l'exercice traitant du numéro atomique, par exemple, il peut choisir les atomes).

5.3. L'historique et l'enseignant

L'enseignant dispose de deux modes pour examiner les travaux effectués. Le premier est le mode *statistique*, où le logiciel fait une synthèse des fichiers historiques de tous les étudiants. Le second, le mode *individuel*, ne concerne qu'un seul étudiant sélectionné par l'enseignant dans une liste nominative. L'enseignant peut consulter l'historique d'un étudiant pour prendre connaissance des travaux réalisés, de l'évolution de l'étudiant par rapport aux différents concepts et des difficultés que l'étudiant a rencontrées.

Ensuite, les mêmes outils sont disponibles dans chacun des deux modes : examen général des travaux réalisés, évolution par rapport à un concept particulier et visualisation des erreurs. L'intérêt d'un examen général des travaux accomplis est de pouvoir cerner plus facilement les tâches où les étudiants ont éprouvé le plus de difficultés et celles qui peuvent être considérées comme acquises. Il est également intéressant pour un enseignant de pouvoir examiner en détail les erreurs commises, afin de mettre à jour d'éventuelles lacunes ou mauvaises compréhensions et d'y remédier pendant les séances de travaux dirigés. La visualisation des erreurs peut se faire de trois manières différentes :

- visualisation linéaire : les erreurs contenues dans l'historique des erreurs sont simplement passées en revue les unes après les autres,
- visualisation quantitative : les exercices sont présentés par nombre d'erreurs décroissant,
- visualisation qualitative : l'enseignant sélectionne l'exercice pour lequel il veut connaître les erreurs qui ont été commises.

Enfin, pour aider l'enseignant à connaître l'évolution d'un étudiant ou de l'ensemble des étudiants pour un concept particulier, un diagramme en bâtons montrant le taux de réussite de la tâche recouvrant ce concept en fonction du temps peut être tracé (figure 16).

6. Une leçon avec ATOMOL

Dans ce paragraphe, nous allons donner la description détaillée d'un problème et simuler son déroulement en compagnie d'ATOMOL. Un des problèmes de niveau premier cycle universitaire auquel nous nous sommes intéressés est la prédiction de la géométrie spatiale d'une molécule par la méthode VSEPR (*Valence Shell Electron Pair Repulsion*) de Gillespie [GIL 70]. Cette méthode repose sur l'étude de l'environnement électronique d'un atome, plus précisément sur la répartition des paires électroniques autour de l'atome, qui tendent à s'orienter de telle sorte que les angles entre les paires soient maximaux. La méthode de Gillespie a l'avantage d'être simple à appliquer tout en pouvant servir à prédire la géométrie de la plupart des

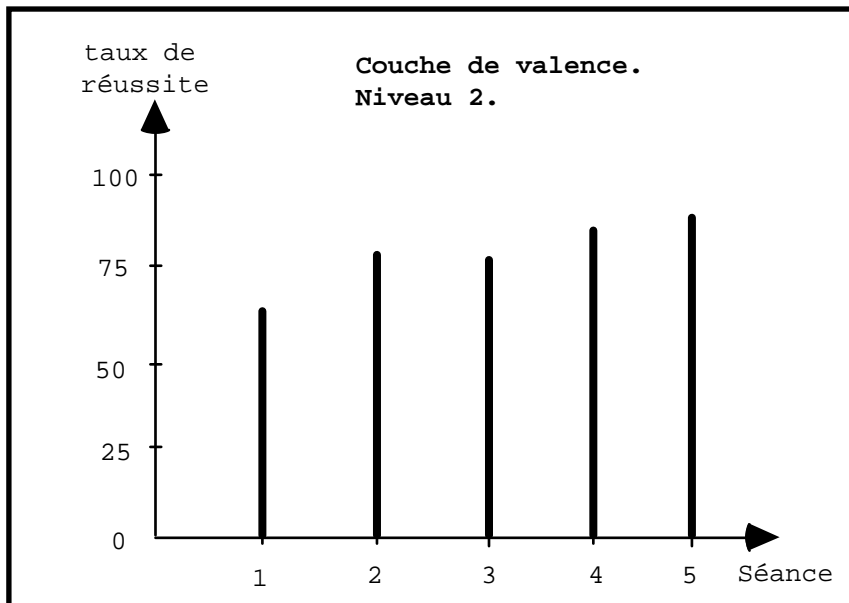


Figure 16. Evolution des étudiants pour le niveau 2 de l'exercice "Couche de valence". Lors de la première séance, le taux de réussite avoisinait les 65%, et a atteint environ 90% à la cinquième séance. Ce diagramme donne à l'enseignant une image explicite de la progression des étudiants.

molécules. Nous nous limitons à l'étude de molécules simples constituées d'un atome dit "central" et de ses voisins directs (figure 17).

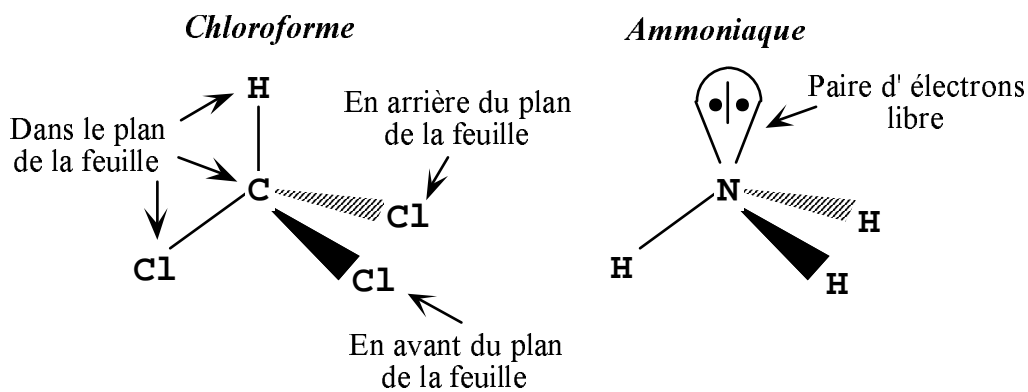


Figure 17. Les molécules de chloroforme (CHCl_3) et d'ammoniaque (NH_3).

6.1. Généralités sur la géométrie d'une molécule

Nous allons nous appuyer sur deux exemples pour présenter quelques concepts essentiels liés à la géométrie d'une molécule : le premier exemple est la molécule de chloroforme CHCl_3 et le deuxième la molécule d'ammoniaque NH_3 (figure 17).

La molécule de chloroforme est constituée d'un atome de carbone central lié à un atome d'hydrogène et à trois atomes de chlore. L'atome de carbone dans la molécule de chloroforme est entouré par quatre liaisons (paires d'électrons liantes) ; la *répulsion maximale* entre les liaisons est obtenue lorsqu'elles s'orientent vers les sommets d'un tétraèdre dont le centre est l'atome de carbone et dont les sommets sont les autres atomes. Ainsi, la géométrie de la molécule de chloroforme est *tétraédrique*.

Dans le cas de la molécule d'ammoniaque, l'environnement de l'atome d'azote central est constitué par trois liaisons avec des atomes d'hydrogènes et d'une paire d'électrons non liante, dite libre. L'atome d'azote est donc entouré par quatre paires d'électrons, dont trois sont liantes et une est libre. La répulsion maximale est à nouveau obtenue lorsque les paires d'électrons s'orientent selon les directions d'un tétraèdre : on dit que la *figure de répulsion* autour de l'atome d'azote est tétraédrique. Cependant, la géométrie de la molécule d'ammoniaque n'est pas tétraédrique, mais possède la forme d'une pyramide trigonale : la paire d'électrons libre est prise en considération pour déterminer la figure de répulsion, puis éliminée pour obtenir la géométrie réelle de la molécule.

6.2. Le problème associé à la méthode de Gillespie

6.2.1. Le sous-problème Structure de Lewis

Pour déterminer la géométrie d'une molécule en utilisant la méthode de Gillespie, il faut connaître auparavant ce qui est appelé la *structure de Lewis* de la molécule. La structure de Lewis d'une molécule précise la répartition des *électrons de valence*, électrons qui sont susceptibles de participer aux liaisons chimiques et qui appartiennent aux couches électroniques périphériques des atomes, appelées *couches de valence*. A son tour, la couche de valence d'un atome est obtenue à partir de la *structure électronique* de l'atome qui indique la répartition des électrons dans différentes couches. Enfin, le *numéro atomique* d'un atome, qui est la position de l'atome dans le tableau périodique des éléments et qui est aussi égal au nombre d'électrons portés par l'atome permet de trouver la structure électronique. En résumé, déterminer la structure de Lewis d'une molécule s'appuie sur les sous-problèmes suivants :

- connaissance du numéro atomique,
- connaissance de la structure électronique,
- reconnaissance de la couche de valence,
- répartition des électrons de valence.

La figure 18 présente l'application de ces différentes étapes dans le cas de la molécule de chloroforme, CHCl_3 .

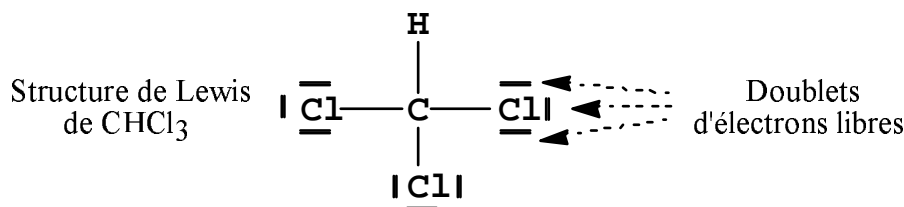
Les numéros atomiques des atomes H, C et Cl sont respectivement 1, 6 et 17. Le nombre d'électrons étant égal au numéro atomique, les atomes H, C et Cl possèdent donc respectivement 1, 6 et 17 électrons.

Pour obtenir la structure électronique fondamentale d'un atome, les électrons sont répartis dans les *orbitales atomiques*, sachant que l'ordre de remplissage est 1s, 2s, 2p, 3s, 3p, que les orbitales de type "s" contiennent au maximum deux électrons et que les orbitales de type "p" contiennent au maximum 6 électrons. Ainsi, la structure électronique fondamentale de l'atome H est $1s^1$ (1 électron dans l'orbitale 1s), celle de l'atome C est $1s^2 2s^2 2p^2$ (2 électrons dans 1s, 2 dans 2s et 2 dans 2p) et celle de l'atome Cl est $1s^2 2s^2 2p^6 3s^2 3p^5$ (2 électrons dans 1s, 2 dans 2s, 6 dans 2p, 2 dans 3s et 5 dans 3p).

La *couche de valence* est obtenue en ne prenant en compte que les électrons situés dans les orbitales de *nombre quantique principal*¹¹ le plus élevé. La couche de valence de l'atome H est donc $1s^1$, celle de l'atome C est $2s^2 2p^2$ et celle de l'atome Cl est $3s^2 3p^5$. Les atomes H, C et Cl possèdent respectivement 1, 4 et 7 électrons de valence.

Ecrire la structure de Lewis consiste alors à préciser la répartition des électrons de valence autour des atomes, sachant qu'une liaison est une mise en commun d'électrons entre deux atomes (chaque atome contribuant pour un électron) et que les électrons qui ne sont pas engagés dans les liaisons forment des doublets d'électrons libres. L'électron de valence de l'atome d'hydrogène est impliqué dans une liaison : l'atome d'hydrogène ne comporte donc pas de doublet d'électrons libre. Il en est de même pour l'atome de carbone dont les 4 électrons de valence participent à des liaisons. En revanche, chacun des atomes de chlore n'est impliqué que dans une seule liaison : les 6 électrons de valence restant forment 3 doublets d'électrons libres.

	Hydrogène H	Carbone C	Chlore Cl
Numéro atomique	1	6	17
Structure électronique fondamentale	$1s^1$	$1s^2 2s^2 2p^2$	$1s^2 2s^2 2p^6 3s^2 3p^5$
Couche de valence	$1s^1$	$2s^2 2p^2$	$3s^2 3p^5$
Nombre d'électrons de valence	1	4	7



¹¹Le nombre quantique principal est le premier nombre quantique qui définit une orbitale. Par exemple, pour l'orbitale 1s, le nombre quantique principal est 1 ; pour l'orbitale 3p il est égal à 3.

Figure 18. *Structure de Lewis de la molécule de chloroforme, CHCl₃.*

6.2.2. Application de la méthode de Gillespie connaissant la structure de Lewis

Une fois la structure de Lewis connue, il reste à compter les *zones de densité électronique* qui définissent des directions particulières autour de l'atome central de la molécule. Les zones de densité électronique sont les paires libres (paires d'électrons périphériques ne participant pas aux liaisons) et les liaisons. Puis, il faut choisir la *figure de répulsion* maximale correspondante. Selon Gillespie, la figure de répulsion est obtenue en tenant compte de toutes les zones de densité électronique, aussi bien des paires libres que des liaisons. Enfin, il s'agit de déterminer la géométrie réelle de la molécule pour laquelle on ne considère que les liaisons et pour laquelle les paires libres ne sont plus prises en compte.

6.2.3. Récapitulatif

La succession des exercices du plus simple au plus complexe pour l'application de la méthode de Gillespie à une molécule est donc la suivante :

- Étape 1 : donner les numéros atomiques de chacun des atomes de la molécule.
- Étape 2 : compléter la structure électronique de chacun des atomes.
- Étape 3 : compléter la couche de valence de chacun des atomes.
- Étape 4 : dessiner la structure de Lewis de la molécule.
- Étape 5 : compter le nombre de zones de densité électronique autour de l'atome central.
- Étape 6 : déterminer la figure de répulsion autour de l'atome central de la molécule
- Étape 7 : déterminer la géométrie de la molécule.

6.2.4. Etude de la géométrie de la molécule CHCl₃ avec ATOMOL

Nous allons examiner en détail l'application de la méthode de Gillespie à la molécule de chloroforme ou CHCl₃ et présenter les exercices tels qu'ils sont proposés par le logiciel en mode de résolution dirigé (cf. § 4.1.).

Étape 1 : le numéro atomique des atomes de la molécule CHCl₃

Les numéros atomiques des atomes H, C et Cl sont respectivement 1, 6 et 17. L'exercice concernant le numéro atomique a déjà été présenté à la figure 12. Cet exercice est une spécialisation de l'exercice générique *Repondre-Au-Clavier* : l'étudiant formule sa réponse en entrant le numéro atomique à l'aide du clavier.

Étape 2 : la structure électronique des atomes de la molécule CHCl₃

La structure électronique fondamentale d'un atome est celle où les électrons sont répartis dans les orbitales atomiques de telle sorte que l'énergie de l'édifice atomique

soit minimale. La structure électronique de l'atome de chlore, qui possède 17 électrons, est $1s^2 2s^2 2p^6 3s^2 3p^5$.

Nous avons imaginé le déroulement suivant pour l'exercice consistant à déterminer la structure électronique fondamentale : les groupes d'orbitales sont présentés dans l'ordre des nombres quantiques croissants, en changeant de ligne chaque fois que le nombre quantique principal augmente d'une unité (figure 19). A chaque groupe d'orbitales est associée une zone rectangulaire qui doit contenir le nombre correct d'électrons. Initialement, toutes les zones contiennent 0 électron, et l'étudiant doit les compléter en choisissant parmi les icônes représentant les nombres de 0 à 10, puis sélectionner les zones dans lesquelles il souhaite placer un nombre d'électrons égal à celui indiqué dans l'icône. Lorsqu'il pense avoir terminé, l'étudiant sélectionne l'icône *FINI*, et la solution qu'il propose est contrôlée par ATOMOL. En cas de succès, l'exercice suivant est proposé à l'étudiant, sinon un message d'erreur s'affiche avec une aide précisant la nature de l'erreur commise.

Quelle est la structure électronique fondamentale de l'atome Cl ?
Placez, dans chaque groupe d'orbitales, le nombre d'électrons approprié.

1s	2						
2s	2	2p	6				
3s	2	3p	5	3d	0		
4s	0	4p	0	4d	0	4f	0
5s	0	5p	0	5d	0	5f	0
6s	0	6p	0	6d	0	6f	0
7s	0						

Control Panel: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0, FINI, AIDE, SOL

Figure 19. Structure électronique fondamentale de l'atome de chlore

Étape 3 : la couche de valence des atomes de la molécule CHCl_3

La couche de valence d'un atome ne contient que ses électrons périphériques, qui sont en général les électrons situés dans les orbitales de nombre quantique principal le plus élevé (figure 18). Le plus souvent, elle se déduit donc facilement de la structure électronique fondamentale. Cependant, la présence de cet exercice est justifiée, puisque dans le cas des atomes dits "éléments de transition" (comme le fer et le cuivre), si n est le nombre quantique principal le plus élevé, la couche de valence contient également des électrons situés dans des orbitales de nombre quantique

principal $n-1$. La forme de l'exercice visant à déterminer la couche de valence est la même que la forme de l'exercice précédent (figure 20).

Quelle est la couche de valence de l'atome Cl ?
Placez, dans chaque groupe d'orbitales, le nombre d'électrons approprié.

1	6
2	7
3	8
4	9
5	10
0	
FINI	
AIDE	
SOL	

1s 0

2s 0 2p 0

3s 2 3p 5 3d 0

4s 0 4p 0 4d 0 4f 0

5s 0 5p 0 5d 0 5f 0

6s 0 6p 0 6d 0 6f 0

7s 0

Figure 20. Couche de valence de l'atome de chlore.

Étape 4 : la structure de Lewis de la molécule CHCl_3

Établir la structure de Lewis d'une molécule revient à dessiner sa formule chimique développée, puis les doublets d'électrons libres que possèdent ses atomes. La structure de Lewis de CHCl_3 a été présentée sur la figure 18. L'exercice associé à la structure de Lewis fait appel à des fonctions qui permettent de sélectionner les atomes et les charges, de tracer les liaisons et les doublets, ainsi qu'à une gomme servant à rectifier d'éventuelles erreurs (figure 21). Ces fonctions de dessin sont communes à d'autres exercices nécessitant également de dessiner des molécules.

Étape 5 : les zones de densité électronique autour de l'atome central dans CHCl_3

Les zones de densité électronique autour d'un atome sont les doublets d'électrons libres et les liaisons. Il est important de connaître le nombre de telles zones autour de l'atome central, puisque l'orientation des liaisons et des doublets d'électrons libres en dépend (c'est-à-dire la figure de répulsion). L'atome de carbone dans la molécule CHCl_3 est lié à quatre atomes et ne possède pas de doublet libre : il y a donc quatre zones de densité électronique autour de l'atome de carbone (figure 18). Le déroulement de l'exercice est identique à celui qui concerne le numéro atomique : l'étudiant formule sa réponse à l'aide du clavier (figure 22).

Structure de Lewis

H

C

Cl

Dessinez la structure de Lewis de CHCl_3 .
Placez les atomes, puis dessinez les liaisons et les doublets.

FINI

AIDE

SOL

Figure 21. Structure de Lewis de CHCl_3 .

SOL

Quel est le nombre de zones de densité électronique définissant une direction autour de C dans CHCl_3 ? Tapez le nombre et validez par RETURN.

Une zone de densité électronique est une paire libre ou une liaison.
Une liaison, qu'elle soit simple ou multiple, ne constitue qu'une seule zone de densité électronique.

Figure 22. Nombre de zones de densité électronique autour de l'atome de carbone dans la molécule CHCl_3 . L'étudiant a demandé de l'aide.

Étape 6 : la figure de répulsion autour de l'atome central dans CHCl_3

Cinq figures de répulsion sont possibles pour des atomes comportant jusqu'à six zones de densité électronique. L'atome de carbone dans la molécule CHCl_3 est entouré par quatre zones de densité électronique : la répulsion maximale dans l'espace est obtenue si les zones prennent les directions des sommets d'un tétraèdre ayant pour centre l'atome de carbone (figure 17). Les différentes figures de répulsion possibles sont proposées à l'étudiant qui doit sélectionner celle qui convient (figure 23).

Chaque fois que l'étudiant se trompe dans son choix, la note est divisée par 2 : cela éloigne la tentation d'essayer chaque bouton.

Quelle est la figure de répulsion définie par les zones de densité électronique autour de C dans CHCl_3 ?

AIDE
SOL

Linéaire Triangle

Tétraèdre Bipyramide trigonale

Octaèdre

Figure 23. Choix de la figure de répulsion de la molécule CHCl_3 .

Étape 7 : la géométrie de la molécule CHCl_3

La géométrie de la molécule s'obtient à partir de la figure de répulsion en ne prenant en compte que les liaisons. Dans le cas de la molécule CHCl_3 , l'atome de carbone ne comporte pas de doublet libre et possède quatre liaisons, sa géométrie est donc tétraédrique (figure 17). L'exercice traitant de la géométrie de la molécule se déroule de la même manière que celui traitant de la figure de répulsion, mais cette fois toutes les géométries qui existent sont affichées (figure 14).

6.2.5. Bilan et variations

Le graphe du problème correspondant à la détermination de la géométrie de la molécule CHCl_3 par la méthode de Gillespie est présenté sur la figure 24. Chaque niveau du graphe est en relation avec une des étapes décrites précédemment.

Voyons maintenant comment s'appliquent les différents modes de résolution à ce problème (cf. § 4.1). Dans le cas du mode de résolution dirigé ou de la reconstruction de problème, les exercices sont proposés en respectant l'ordre des étapes successives : exercice 1, exercice 2, exercice 3, puis exercice 4, exercice 5, exercice 6, et ainsi de suite jusqu'au dernier exercice, exercice 13.

Si le mode de résolution est le retour aux prérequis, le parcours du graphe dépend des actions de l'étudiant. Supposons que l'étudiant n'ait pas su déterminer directement la géométrie de la molécule (exercice 13) : il est renvoyé à la figure de répulsion (exercice 12). Supposons encore qu'il n'ait pas su répondre à cet exercice, ni à celui concernant le nombre de zones de densité électronique autour de l'atome de carbone (exercice 11). Il est donc renvoyé à l'exercice concernant la structure de Lewis de CHCl_3 (exercice 10). En cas de succès, l'étudiant retourne successivement à l'exercice 11, puis l'exercice 12 et finalement l'exercice 13.

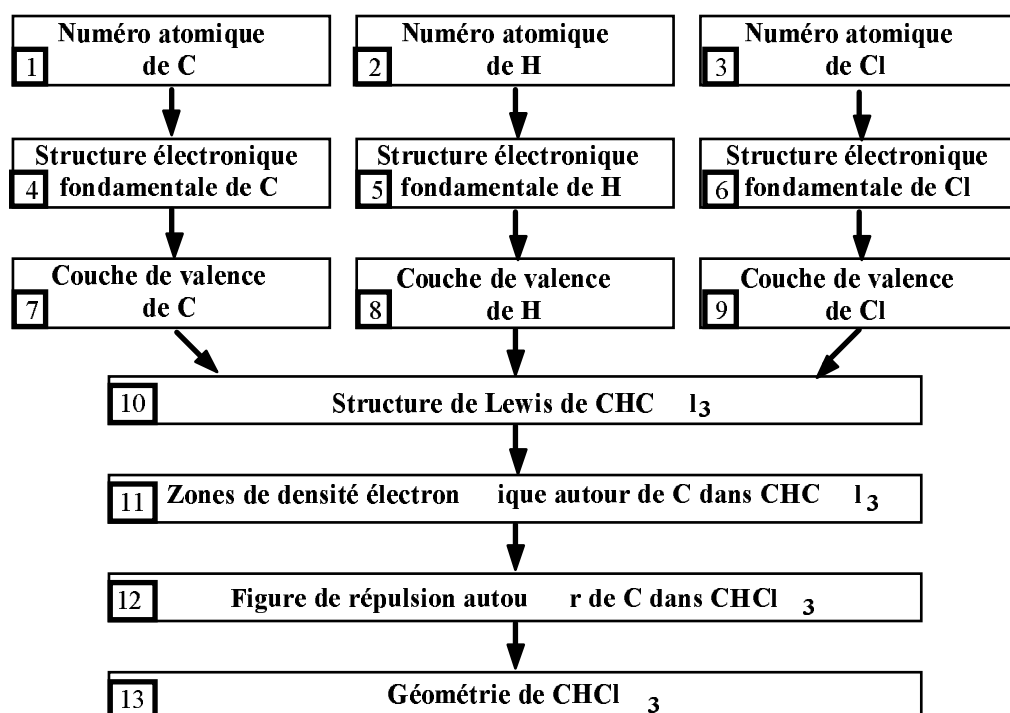


Figure 24. Problème visant à déterminer la géométrie de la molécule CHCl_3 .

7. Discussion

Dans un premier temps, nous montrons que l'approche adoptée pour la réalisation du logiciel ATOMOL satisfait aux exigences pédagogiques attendues, dans le domaine traité. Nous discutons ensuite des qualités que le logiciel présente du point de vue de l'architecture, puis de celui de l'interactivité.

7.1. Le point de vue pédagogique

Parmi les tuteurs, il est possible de distinguer deux grandes familles. La première comprend les tuteurs intelligents construits autour d'un système qui permet de résoudre des problèmes dans le domaine étudié. En chimie, il existe de tels tuteurs par exemple pour l'aide à la détermination des produits de réactions inorganiques [BIR 92] ou la conversion du nom IUPAC d'une molécule en sa formule chimique [EJM 93]. Ces systèmes ont la capacité de conserver la trace de leur raisonnement, et ainsi d'en livrer les différentes étapes. Cependant, la seule trace du raisonnement utilisé par le système expert pour résoudre un problème n'est pas toujours une explication exploitable. L'auteur d'AMALIA, système destiné à l'apprentissage des manipulations d'expressions algébriques [VIV 87], souligne que la trace du raisonnement est suffisante pour *commenter* une solution, mais non pour *expliquer* ou *justifier* une solution. Dans AMALIA, deux systèmes cohabitent : un système de résolution de problèmes et un tuteur. Le système de résolution résout des problèmes

de manipulation de fonctions numériques ; le tuteur met en œuvre des règles particulières utiles pour atteindre un objectif pédagogique fixé.

Dans une deuxième famille de tuteurs, les connaissances du domaine à enseigner sont décomposées en unités pédagogiques élémentaires, appelées *items d'évaluation* [GAV 91] ou *salles pédagogiques* [REG 88]. A chaque unité est associé un des concepts que l'apprenant doit maîtriser. Dans chacune des unités sont également définis des liens utilisés par le tuteur pour gérer le déroulement de l'apprentissage, notamment dans choix des tâches adaptées à l'apprenant.

Le système ATOMOL, qui ne dispose pas de “module de raisonnement” autonome, s'apparente aux systèmes de la deuxième famille. Les unités pédagogiques sont les exercices, chacun d'eux traitant indépendamment d'un concept élémentaire de chimie. Cette approche convient d'autant mieux qu'un concept élémentaire de chimie ne fait en général intervenir qu'un petit nombre de règles rappelées sous forme d'aide dans l'exercice associé ; les aides suffisent donc en principe pour résoudre l'exercice. Le déroulement d'un problème est ordonné par la relation “*est un prérequis de*” qui est l'unique lien entre les exercices qui composent le problème. Le déroulement global de l'apprentissage est contrôlé par la progression souhaitée par l'enseignant, qui définit des liens d'antériorité entre les tâches.

7.2. Le point de vue de l'architecture

Le module d'enseignement et la représentation de l'expertise chimique sont programmés dans un formalisme objet, qui autorise des extensions, sans que le travail déjà réalisé ne soit remis en cause mais au contraire réutilisé, ce qui ne peut que favoriser l'évolution du logiciel. D'abord, si les connaissances ou les capacités de raisonnement du module expert deviennent insuffisantes, il est possible d'enrichir les connaissances et les objets chimiques avec de nouvelles propriétés. Ensuite, le logiciel ATOMOL dispose d'un module d'enseignement générique sur lequel peuvent être greffées de nouvelles tâches spécifiques. Le module d'enseignement repose en effet sur un ensemble d'objets organisé en une hiérarchie d'héritage. Les objets situés dans la partie supérieure de la hiérarchie sont des tâches génériques. Chaque tâche générique décrit la forme particulière d'une tâche, en relation avec le déroulement et la présentation de la tâche lorsqu'elle est proposée à l'étudiant. Le niveau générique du module d'enseignement est indépendant du domaine enseigné, et il est donc *réutilisable* pour des domaines autres que celui de la chimie¹². Les objets situés dans la partie inférieure de la hiérarchie sont spécifiques au domaine de la chimie. Une tâche de chimie hérite la forme d'une tâche générique, mais définit le fond qui lui est associé : l'énoncé, les aides, le traitement des erreurs et la formulation de la solution. La création de nouvelles tâches de chimie est donc facilitée, puisque seul le fond doit être spécifié, la forme étant héritée.

Toutefois, ajouter de nouvelles tâches sur la base de l'environnement générique d'ATOMOL reste un travail à réaliser, alors que certains systèmes rendent accessible

¹²Nous avons par exemple programmé quelques exercices d'orthographe sur la base des exercices génériques existants.

la réalisation de tuteurs intelligents à des enseignants. Deux exemples sont GAMETE (Générateur Automatique de systèMes d'apprEntissage auTonomE [REG 88], et S.G.T.I. (Système Générateur de Tuteur Intelligent) [BEN 90]. Cependant, ces systèmes sont encore des prototypes, et la représentation de l'expertise en chimie qui a été réalisée pour ATOMOL est trop complexe pour être intégrée et exploitée par l'un de ces systèmes. De plus, l'interface d'ATOMOL a été conçue de manière à permettre la saisie et la visualisation des molécules dans les exercices, ce qui n'est pas le cas des interfaces de GAMETE et S.G.T.I..

7.3. L'interactivité

Le dialogue entre l'étudiant et le programme se fait au travers d'une interface conviviale et homogène. L'étudiant effectue la plupart de ses actions à l'aide de la souris. Deux modes d'interaction sont possibles : l'étudiant peut se laisser guider complètement pendant une séance de travail, ou au contraire choisir lui-même les sujets qui l'intéressent. Quand le choix des tâches est laissé au logiciel, le passé de l'étudiant est pris en considération : le choix d'une tâche est alors fonction des acquis, des difficultés et des objectifs à atteindre par l'étudiant. Plusieurs modes de résolution de problème sont également possibles : soit le problème est dirigé de façon préétablie, soit l'étudiant reconstruit la démarche du problème, soit l'étudiant est renvoyé à des prérequis en cas d'erreur. Les différents modes d'interaction et de résolution de problème permettent ainsi à l'étudiant d'adapter le déroulement des séances d'apprentissage à son caractère, son niveau de connaissance et son intérêt.

Un aspect intéressant dans le dialogue avec le logiciel ATOMOL est la possibilité qui est donnée à l'étudiant de formuler une réponse en dessinant une molécule telle qu'il la dessinerait habituellement sur un papier : les représentations planes des molécules font partie intégrante du langage du chimiste. Un logiciel comme ATOMOL ne serait pas envisageable sans de telles possibilités graphiques. En effet, le dessin des structures moléculaires est interprété en terme d'objets chimiques, atomes, liaisons et molécule. Les caractéristiques de la molécule sont obtenues par les fonctions de perception chimique qui lui sont associées.

La présence d'une représentation de la molécule permet un mode d'interaction encore plus souple que ceux qui existent déjà, à savoir laisser l'étudiant poser lui-même des questions. Nous pourrions, par exemple, laisser l'étudiant dessiner la molécule sur laquelle il veut travailler, plutôt que de lui imposer des molécules prédéfinies. Nous avons envisagé cette possibilité et nous l'avons écartée après avoir considéré le public auquel est destiné le logiciel. Ce sont des étudiants débutants en chimie, qui doivent assimiler de nouvelles connaissances, et qui ne possèdent pas suffisamment de recul pour juger de la pertinence des informations qui leur sont transmises. Nous avons estimé que chaque tâche proposée à l'étudiant par le logiciel doit être contrôlée préalablement par un enseignant, afin qu'aucun résultat erroné ne soit jamais communiqué à l'étudiant. Nous avons imposé cette condition après avoir examiné le problème de la réaction chimique. Même les systèmes de synthèse assistée par ordinateur les plus sophistiqués, qui disposent d'une base de réactions importante, ne parviennent pas à prendre en considération tous les paramètres qui interviennent

lors de réactions chimiques ; la validité des résultats obtenus doit être confirmée par un chimiste expérimenté. Dans le cas de l'enseignement de la réaction chimique, sujet que nous allons aborder, il est préférable de limiter le domaine d'exploration. Ensuite, s'il incombe à l'étudiant de poser les questions, il est rare qu'il soit capable d'aborder les divers aspects recouverts par un concept et de formuler correctement les questions.

8. Conclusion

Dans cet article, nous avons présenté les aspects essentiels qui ont accompagné la réalisation du logiciel d'enseignement assisté par ordinateur ATOMOL. Après avoir décrit l'architecture du logiciel, nous avons examiné les principaux modules qui le composent, à savoir le module expert, le module d'enseignement et l'historique de l'étudiant. Le module expert qui contient l'expertise chimique nécessaire au logiciel est construit autour d'une représentation à objets de la molécule. Le module d'enseignement, conçu également dans un formalisme objet, contient la description des tâches proposées à l'étudiant. L'historique de l'étudiant contient des informations relatives au travail de l'étudiant, qui sont utilisées par le module d'enseignement pour adapter l'enseignement à un étudiant particulier et qui peuvent être consultées par l'enseignant. Nous avons ensuite présenté le déroulement complet d'un problème : la détermination de la géométrie d'une molécule par la méthode VSEPR de Gillespie. Nous avons terminé par une discussion relative aux aspects pédagogiques, informatiques et à l'interactivité du logiciel. Les nombreux extraits de code présentés font également de ce texte un exemple de conception de système de représentation à base d'objets.

Les premiers intéressés par le programme ATOMOL sont les étudiants, auxquels le logiciel donne la possibilité de s'entraîner sur des questions essentielles de chimie (questions portant sur la structure des atomes et des molécules), en complément des travaux dirigés classiques. Les problèmes proposés par ATOMOL concernent l'écriture de la formule de Lewis des molécules et la détermination de la géométrie des molécules par la méthode de Gillespie. Ces problèmes font appel à sept exercices, qui en constituent les étapes élémentaires. Actuellement, les exercices portent sur les 54 premiers atomes du tableau périodique (les cinq premières périodes) ; l'enseignant a la possibilité de créer des exercices avec d'autres atomes, puisque tous les atomes du tableau périodique sont représentés dans le module expert. De même, l'enseignant peut enrichir la base de molécules, qui contient actuellement un échantillon d'une cinquantaine de molécules.

L'enseignant, quant à lui, peut suivre le travail des étudiants, grâce aux informations contenues dans les fichiers historiques qui décrivent le passé des étudiants. D'autres fonctionnalités du logiciel n'ont pas été décrites ici : ainsi, l'enseignant peut définir la progression des tâches qui est utilisée lorsque l'étudiant se laisse guider par le programme ; il dispose également d'outils permettant d'enrichir les exercices avec de nouveaux exemples, de les modifier ou de les supprimer, et de composer des problèmes.

Dans l'immédiat, nous comptons ajouter au logiciel ATOMOL un nouveau problème qui permet de faire le lien entre la géométrie des molécules et la théorie des orbitales moléculaires. Nous comptons aussi développer un nouveau logiciel, nommé ISOMOL, à partir de l'environnement générique d'ATOMOL et en profitant de fonctions de perception chimique qui ont déjà été programmées mais ne sont pas exploitées dans ATOMOL [FIS 93]. Le logiciel ISOMOL abordera des problèmes de stéréochimie, comme l'attribution de la configuration des stéréocentres et la stéréochimie des liaisons doubles ; il amènera également l'étudiant à manipuler les différentes représentations des molécules. Ensuite, nous souhaitons aborder la réaction chimique, ce qui nécessite la programmation d'une connaissance supplémentaire importante, qui comprend notamment une représentation des réactions. En revanche, la représentation des molécules reste parfaitement réutilisable dans le contexte de la réaction chimique, puisqu'elle a été conçue parallèlement pour un système de synthèse assistée par ordinateur.

Enfin, le logiciel ATOMOL, qui est opérationnel depuis peu, n'a encore pu être suffisamment évalué. Pour le moment, nous n'avons pu laisser les étudiants tester le logiciel que pendant une période très courte, et nous n'avons pas recueilli suffisamment leurs remarques, appréciations et commentaires. Cependant, une évaluation du logiciel devrait avoir lieu dans plusieurs sites durant l'année universitaire 94-95.

Pour terminer, nous précisons quelques détails techniques à l'attention des enseignants et chercheurs qui seraient intéressés par une version du logiciel lorsqu'une telle version sera disponible. Le logiciel ATOMOL a été réalisé et fonctionne sur un micro-ordinateur Macintosh équipé d'un disque dur et disposant de trois mégaoctets de mémoire centrale (pour un confort d'utilisation maximal, il est recommandé d'utiliser un ordinateur de type Macintosh II). La taille de l'application est d'environ 1,8 mégaoctets. L'application est accompagnée de plusieurs fichiers renfermant les molécules, les données pour les tâches, les dessins utilisés dans la présentation des exercices et les historiques des étudiants. Deux langages de programmation ont été utilisés, Macintosh Common Lisp et YCHEM qui, comme nous l'avons dit, est un langage à objets écrit en Lisp.

9. Bibliographie

- [AAF+ 91] ADULDECHA S., AKHTER P., FIELD P., NAGLE P., O'SULLIVAN E., CONNOR K. O., HATHAWAY B.: *The Use of the Desktop Molecular Modeller Software in the Teaching of Structural Chemistry*, Journal of Chemical Education, 68(7) : 576-583, 1991.
- [AD 93] AZAY P., DUMONT C.: *Contrôle d'installations de laboratoire à l'aide d'un micro-ordinateur en fonctionnement multitâche*, Actes des sixièmes Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Conservatoire National des Arts et Métiers, Paris, pages 19-21, 1-3 avril 1993.
- [AH 75] ASH J. E. AND HYDE E.: *Chemical Information Systems*, Ellis Horwood, Chichester, 1975.

- [AP 90] ADAM-NICOLLE A., PORQUET C.: *Object-Based Representations and Object-Oriented Languages : Possible Applications in Chemistry*, New Journal of Chemistry, 14 : 905-911, 1990.
- [ARN 83] ARNAUD P.: *Cours de chimie organique*, Gauthier-Villars, Paris, 1983.
- [BEN 90] BENSLIMANE A.: *Modèles pour un générateur de tuteurs intelligents*, Thèse de l'Université de Clermont II, 1990.
- [BIR 92] BIRK J. P.: *The Computer as Student. An Application of Artificial Intelligence*, Journal of Chemical Education, 69(4) : 294-295, 1992.
- [BMA 84] BARONE R., MEYER R., ARBELOT M.: *R/S : Apple Stereochemistry Program*, Journal of Chemical Education, 61(6) : 524-525, 1984.
- [BUR 82] BURTON R. R.: *Diagnosing bugs in a simple procedural skill*, in D. Sleeman and J. S. Brown, "Intelligent Tutoring Systems", Academic Press, London, pages 157-183, 1982.
- [CCC 86] CABROL D., CACHET C., CORNELIUS R.: *An Heuristic Problem Solver : GEORGE*, Computers in Education, 10(1) : 81-87, 1986.
- [CLE 89] CLEMENT R.: *ORGAVISU : logiciel d'aide à l'apprentissage de la nomenclature et des structures moléculaires en chimie organique*, Actes des quatrième Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Université de Pau, pages 66-67, 11-13 septembre 1989.
- [COR+ 88] CLAES G., OUNIS O., RAZOARIVELO Z., SALEMBRIER P., SRIDHARANM. S.: *Starguide : un générateur de systèmes d'autoformation à l'usage des logiciels*, Technique et Science Informatiques, 7(1) : 69-78, 1988.
- [CRF 88] CABROL D., RABINE J. P., FOREST T. P.: *An Educational Problem Solving Partner in Prolog for Learning Infrared Spectroscopic Analysis*, Computers in Education, 12(1) : 241-246, 1988.
- [CRR+ 91] CABROL D., RABINE J. P., RICARD D., ROUILLARD M., FORREST T. P.: *EXP'AIR : un logiciel pour l'apprentissage de l'interprétation des données de spectroscopie infra-rouge*, pages 123-148, dans [QUE 91].
- [DDG 89] DUMONT B., DROUHARD J. P., GRUGEON B.: *FRACT : un exemple d'apports possibles de l'intelligence artificielle à la pédagogie de l'erreur en Mathématiques*, Journées "EIAO" du PRC-GDR Intelligence Artificielle, ENS de Cachan, pages 191-200, 1989.
- [DDR 89] DUMON A., DUBREUIL F., RABINE J. P.: *VSEPR 1 : un dialogue pour la prévision des structures géométriques de base des molécules et ions moléculaires*, Actes des quatrième Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Université de Pau, page 72, 11-13 septembre 1989.
- [DED 86] DEDE C.: *A review and synthesis of recent research in intelligent computer-assisted instruction*, International Journal of Man-Machine Studies, 24 : 329-353, 1986.
- [DLL 89] DIAWARA B., LALLEMANT M., LARPIN J. P.: *CRISTAL : logiciel de représentation en 3D de structures cristallographiques pour l'autoformation des étudiants de premier et second cycles universitaires*, Actes des quatrième Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Université de Pau, pages 36-40, 11-13 septembre 1989.
- [DUC 88] DUCOURNAU R.: *YAFOOL. Version 3.22. Manuel de référence*, SEMA GROUP, Montrouge, 1988.
- [EJM 93] EGGERT A. A., JACOB A. T., MIDDLECAMP C. H.: *Converting Chemical Names to Formulas: A Second Expert Problem*, Journal of Chemical Information and Computer Science, 33(3) : 458-465, 1993.
- [FIS 93] FISCHER D.: *Enseignement Assisté par Ordinateur : Conception d'un Logiciel d'Assistance aux Travaux Dirigés de Chimie en DEUG*, Thèse de l'Université de Strasbourg I, 1993.
- [GAV 91] GAVIGNET E.: *Environnement de conception de systèmes d'apprentissage : Une modélisation de la connaissance pédagogique*, Thèse de l'Université de Nancy I, 1991.

- [GIL 70] GILLESPIE R. J.: *The Electron-Pair Repulsion Model for Molecular Geometry*, Journal of Chemical Education, 57(1) : 18-23, 1970.
- [HOL 84] HOLDSWORTH D. K.: *Chromatography: Use of a Microcomputer to Introduce Laboratory Techniques*, Journal of Chemical Education, 61(9) : 790-791, 1984.
- [MNC+ 89] MASINI G., NAPOLI A., COLNET D., LEONARD D., TOMBRE K.: *Les Langages à Objets*, InterEditions, Paris, 1989.
- [MSA 80] MOORE C., SMITH S., AVNER R.A.: *Facilitation of Laboratory Performance Through CAI*, Journal of Chemical Education, 57(3) : 196-198, 1980.
- [NAP 90] NAPOLI A.: *Using Frame-Based Representation Languages to Describe Chemical Objects*, New Journal of Chemistry, 14(12) : 913-919, 1990.
- [NAP 92] NAPOLI A.: *Représentations à objets et raisonnement par classification en intelligence artificielle*, Thèse d'Etat, Université de Nancy I, 1992.
- [NL 93] NAPOLI A., LAURENÇO C.: *Représentations à objets et classification. Conception d'un système d'aide à la planification de synthèses organiques*, Revue d'intelligence artificielle, 7(2) : 175-221, 1993.
- [NAU 89] NAUDIN A.: *Nomenclature en chimie organique*, Actes des quatrièmes Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Université de Pau, pages 64-65, 11-13 septembre 1989.
- [NIL 80] NILSSON N. J.: *Principles of Artificial Intelligence*, TIOGA Publishing Company, Palo Alto, California, 1980.
- [NV 88] NICAUD J. F., VIVET M.: *Les tuteurs intelligents: réalisations et tendances de recherches*, Technique et Science Informatiques, 7(1) : 23-45, 1988.
- [PLL 93] PERCHE A., LHERBIER S., LOOCK R.: *Simulation pré-laboratoire en DEUG 1^o année de chimie. Titrages potentiométriques. Le logiciel TITRAGES*, Actes des sixièmes Journées sur les Méthodes Informatiques dans l'Enseignement de la Chimie, Conservatoire National des Arts et Métiers, Paris, pages 90-91, 1-3 avril 1993.
- [QUE 91] QUERE M.: *Systèmes experts et enseignement assisté par ordinateur*, OPHRYS, 1991.
- [REG 88] REGOURD J. P.: *GAMETE: un système auteur pour l'EIAO*, Technique et Science Informatiques, 7(1) : 59-68, 1988.
- [RRC+ 88] RABINE J.P., ROUILLARD M., CABROL D. DUBREUIL F.: *A N.M.R. Spectrum Analysis: a Problem Solving Partner Based on the Student's Approach*, Computers in Education, F. Lovis and E. D. Tagg (eds.) Elsevier Science Publishers B. V. (North-Holland) IFIP 1988.
- [SEL 87] SELF J.: *Student models: What use are they?*, IFIP/TC3, FRASCATI, May 1987, North Holland (R. Lewis, P. Ercoli eds.).
- [SET 87] SETTLE F.A.: *The Application of Expert Systems in the General Chemistry Laboratory*, Journal of Chemical Education, 64(4) : 340-345, 1987.
- [VIV 87] VIVET M.: *Systèmes experts pour enseigner: meta-connaissances et explications*, ACTES COGNITIVA 87, Paris, La Vilette, 3-8, mai 1987.

Dominique FISCHER a soutenu sa thèse de chimie informatique et théorique à l'Université Louis Pasteur de Strasbourg, en 1993. Il a préparé sa thèse au laboratoire de Modèles Informatiques Appliqués à la Synthèse. Dans ce cadre, il a travaillé sur une représentation à base d'objets des concepts de chimie organique, a développé des algorithmes de perception informatique des caractéristiques structurales et chimiques des molécules, puis a conçu un logiciel d'enseignement assisté par ordinateur pour la chimie. Après sa thèse, il a poursuivi la réalisation du logiciel d'enseignement au Centre d'AutoFormation et d'Ingénierie Educative, toujours à Strasbourg. Dominique

Fischer s'intéresse également à l'enseignement traditionnel, puisque, depuis six ans, il enseigne les mathématiques dans un collège (Tél. : 88 41 66 78)

Michèle KIRCH est maître de conférences de chimie à l'Université Louis Pasteur de Strasbourg et chargée de mission auprès de la Vice Présidence aux enseignements pour le développement de l'enseignement multi-média. A travaillé sous la direction du Prof. Jean-Marie Lehn de 1971 à 1983 (modèle de transport ionique à travers des membranes artificielles ; photoréduction de l'eau). A rejoint en 1984 le Groupe d'Etudes et de Recherches sur les Sciences de l'ULP (GERSULP) dirigé par le Professeur Baudoin Jurdant, où elle travaille sur des questions concernant le langage de la chimie et le contenu et les méthodes de l'enseignement de la chimie en 1^{er} cycle universitaire. A été le maître d'œuvre de la création par l'ULP en 1992 du Centre d'Autoformation et d'Ingénierie Educative (Tél. 88 41 61 10).

Amedeo NAPOLI est chercheur CNRS au Centre de Recherche en Informatique de Nancy (CRIN CNRS - INRIA Lorraine). Ses domaines d'activités principaux sont en relation avec la représentation de connaissances et l'étude des modes de raisonnement. Il s'intéresse plus particulièrement aux représentations à base d'objets et au raisonnement par classification. Du côté des applications, il collabore depuis longtemps avec des chimistes sur la conception de systèmes à base de connaissances pour la planification de synthèses organiques (en particulier dans le cadre du GDR CNRS 1093). Son activité dans le domaine des sciences de l'éducation reste liée au domaine d'application qu'est la chimie organique (Tél. 83 59 20 68).