

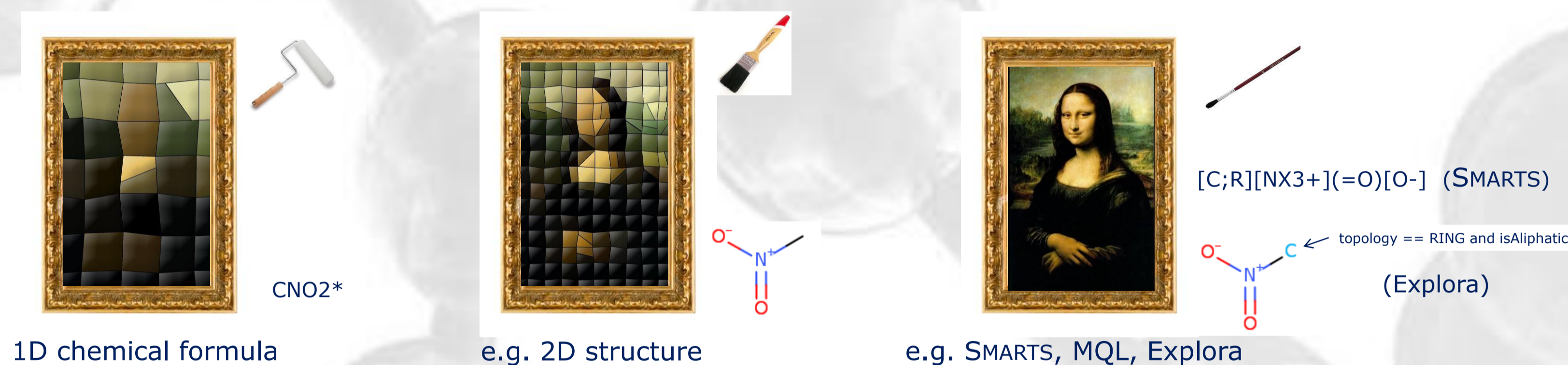
## Abstract

Exploring chemical structure spaces is an increasingly important activity in modern Bio/Chemo-informatics applications. Structural queries in chemical databases, fragment statistics, patenting and SAR modeling are only a few examples of where structural spaces and queries are key elements. At Lhasa Limited we use structural queries to define 2D toxicophores (Derek<sup>1</sup>) and biotransformation sites (Meteor<sup>2</sup>); they capture the expert's SAR knowledge in these fields in the form of structural queries. From an abstract view point, structural queries describe a domain in the chemical space that is of interest in a given context. In most contexts, the accuracy of these queries is critical and their fine grained expression is challenging.

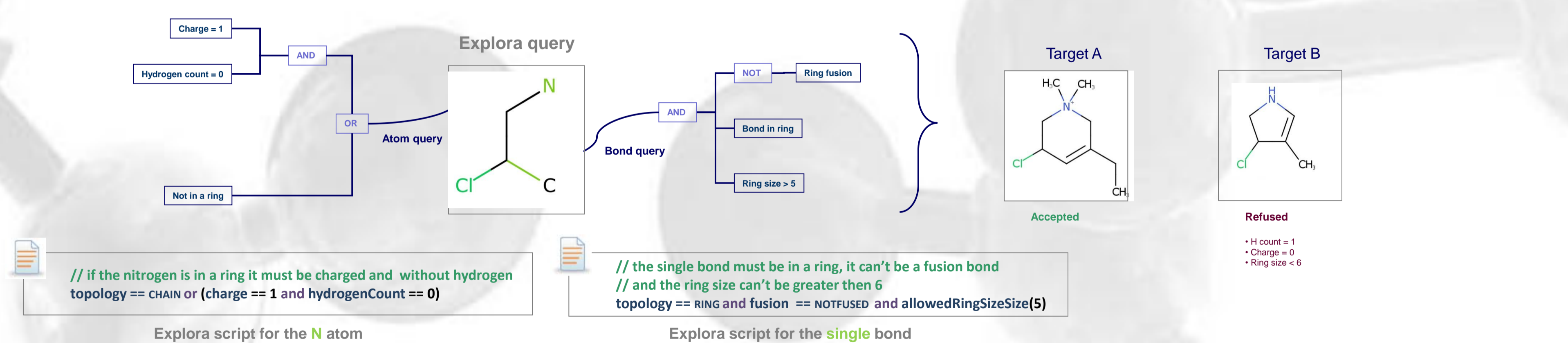
Current approaches use structural information refined with generic atoms or bonds or more sophisticated descriptors like SLN<sup>3</sup>, SMARTS<sup>4</sup> strings and more recently MQL<sup>5</sup> queries. In some cases, these approaches are not flexible enough and the resulting queries are often not intuitive. In order to overcome these limitations and create accurate and easy to read chemical scopes, we have developed Explora, a powerful structural query language. This work is an extension of the development of the concept of L-Patterns (logical Markush structures)<sup>6</sup>.

## Defining accurate chemical scopes

Describing a chemical scope or a structural query is like creating a painting, the language used to define this scope is like the brush used by the painter. The finer the brush the more accurate will be the painting. If the scientist (the painter) does not have the right query language (the right brush) he won't be able to accurately define the intended chemical scope.



Explora is an expression language that is used to define constraints on individual atoms, bonds or the structure as a whole. We wanted to keep the benefit of a 2D visual representation of the structure together with the power of an expression based attribute definition. The user defines a query structure using a structure editor. Each atom, bond or the structure as a whole can then be further enriched with an Explora script to express a set of constraints. Explora scripts using descriptors, functions and operators allow the user to build sophisticated constraints thus contouring accurately the intended chemical scope of the query. The example below presents a simple use case of Explora.



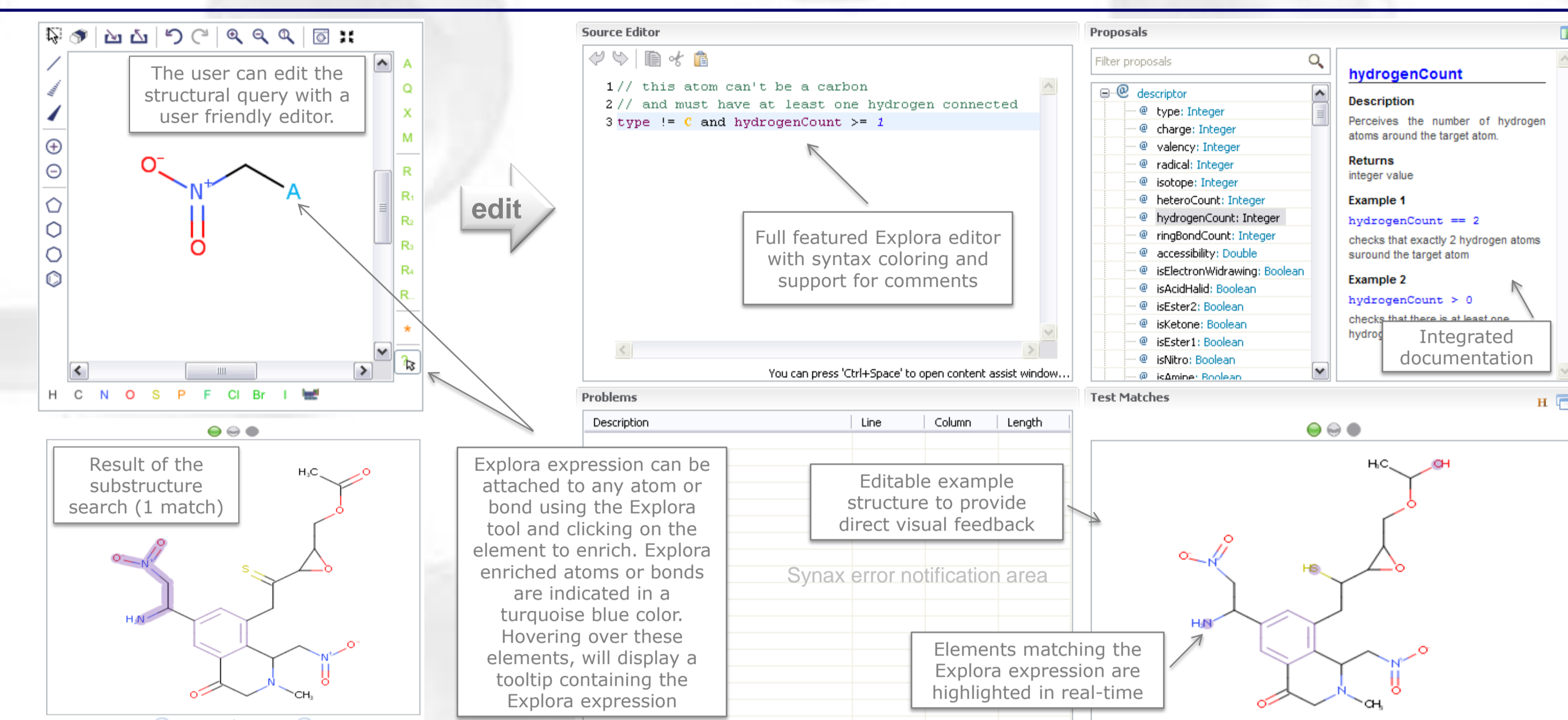
## An intuitive scripting language

Ideally we would like Explora scripts to be close to plain English statements. We therefore defined a context-free grammar that supports functions, operators, lists, ranges, etc. and introduces the concept of descriptor (function with zero arguments). The grammar has been designed to provide high legibility and minimum interpretation effort. Since Explora is meant to be extended to a full chemical scripting language in the future, its grammar has also been carefully shaped to anticipate features like variables, value assignments, flow control, encapsulation, etc. The examples in the following tables demonstrate the flexibility and the legibility of Explora.

Atom queries	Explora	Bond queries	Explora	Structure queries	Explora
Charged N or O	<chem>type == {N,O} and charge != 0</chem>	Double or aromatic ring bond	<chem>type == {DOUBLE, AROMATIC} and topology == RING</chem>	Molecular weight less than 500 daltons	<chem>molecularWeight &lt; 500</chem>
Atom in ring of size 5 or 6 with no H's	<chem>allowedRingSize(5,6) and hydrogenCount == 0</chem>	Heterocyclic ring bond	<chem>ringType(HETEROCYCLIC)</chem>	Number of heavy atoms between 30 and 50	<chem>atomCount == [30,50]</chem>
Spatial accessibility greater than 0,8	<chem>accessibility &gt; 0.8</chem>	Single or fused bond	<chem>type == {SINGLE} or fusion == FUSED</chem>	logP larger than 3	<chem>logP &gt; 3</chem>
Atom with less than 2 hetero atom neighbours	<chem>nbHetero &lt; 2</chem>	Bond in a ring of size 5	<chem>requiredRingSize(5)</chem>	Number of O or S atoms beyond the matched motif must be less than 2	<chem>offPathAtom(O,S) &lt; 2</chem>
Cation	<chem>charge &gt; 0</chem>	Fusion bond between 2 aliphatic rings	<chem>fusion == ALIPHATIC_ALIPHATIC</chem>		

## User friendly

To facilitate the input of Explora based structural queries we have developed a structure editor and a fully featured Explora editor. The user can easily attach Explora scripts to an atom or a bond using the Explora tool in the structure editor. The Explora editor supports syntax colouring, error highlighting, multiline comments, etc. It also provides integrated help and the possibility to apply in real-time the Explora script on a test structure.



The user can edit the structural query with a user friendly editor.

Full featured Explora editor with syntax coloring and support for comments

Explora expression can be attached to any atom or bond using the Explora tool and clicking on the element to enrich. Explora enriched atoms or bonds are indicated in a turquoise blue color. Hovering over these elements, will display a tooltip containing the Explora expression

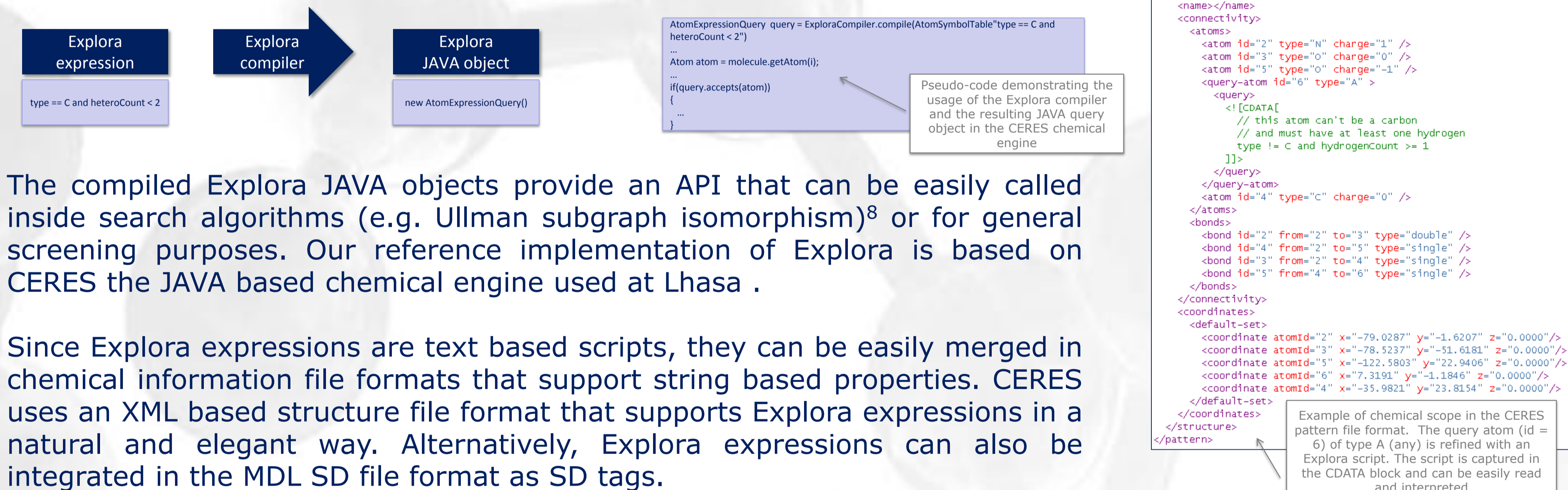
Editable example structure to provide direct visual feedback

Syntax error notification area

Elements matching the Explora expression are highlighted in real-time

## Fast and Easy to integrate

The Explora grammar has been designed using ANTLR<sup>7</sup> and we developed the corresponding compiler in JAVA. Explora scripts are compiled (at JAVA runtime) directly into new instances of JAVA objects ready to be invoked in a query (at a later stage). They are therefore very fast and easy to integrate in any software running on the JAVA platform. A typical Explora expression is executed in a few micro seconds.



Explora expression: type == C and heteroCount < 2

Explora compiler

Explora JAVA object: new AtomExpressionQuery()

Pseudo-code demonstrating the usage of the Explora compiler and the resulting JAVA query object in the CERES chemical engine

```
AtomExpressionQuery query = ExploraCompiler.compile(AtomSymbolTable type == C and heteroCount < 2);
Atom atom = molecule.getAtom(i);
if(query.accept(atom)) {
    ...
}
```

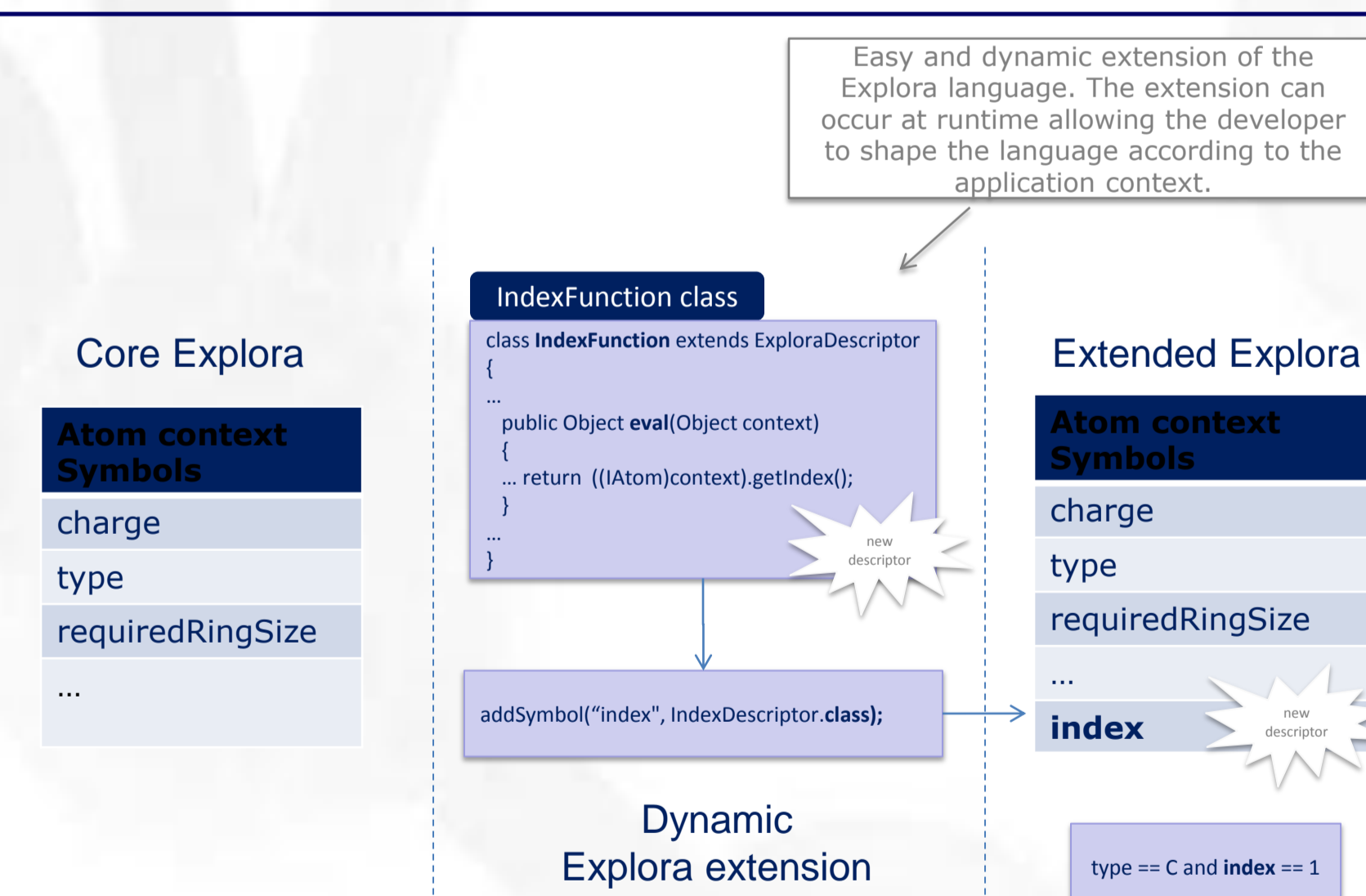
Example of chemical scope in the CERES pattern file format. The query atom (id = 6) of type A (any) is refined with an Explora script. The script is captured in the CDATA block and can be easily read and interpreted

```
<pattern logic="and({})" xmlns="urn:org.lhasa:limited:ceres">
  <name>Explora example</name>
  <structure format="ceres">
    <name></name>
    <connectivity>
      <atoms>
        <atom id="2" type="N" charge="1" />
        <atom id="3" type="O" charge="0" />
        <atom id="5" type="O" charge="-1" />
        <query-atom id="6" type="A" />
      </atoms>
      <CDATA[
        // this atom can't be a carbon
        // and must have at least one hydrogen
        type = C and hydrogenCount >= 1
      ]>
    </query>
    </query-atom>
    <atom id="4" type="C" charge="0" />
  </atoms>
  <bonds>
    <bond id="2" from="2" to="3" type="double" />
    <bond id="3" from="2" to="5" type="single" />
    <bond id="4" from="2" to="4" type="single" />
    <bond id="5" from="4" to="6" type="single" />
  </bonds>
  </connectivity>
  <coordinates>
    <default-set>
      <coordinate atomid="2" x="-79.0287" y="1.8307" z="0.0000" />
      <coordinate atomid="3" x="-79.5297" y="-12.6181" z="0.0000" />
      <coordinate atomid="5" x="-122.5803" y="22.9406" z="0.0000" />
      <coordinate atomid="6" x="7.3193" y="1.1846" z="0.0000" />
      <coordinate atomid="4" x="31.9851" y="73.8381" z="0.0000" />
    </default-set>
  </coordinates>
</structure>
</pattern>
```

## Extensible

The current version of Explora provides a collection of ready-to-use operators, functions and descriptors designed to cover our needs in SAR modeling. The language can be easily extended with new functionalities written in Java thanks to a simple and dynamic language definition API. The developer creates a JAVA class that implements the extension API and implements the desired functionality in the eval() method. An instance of this class can then be registered at run time as a new feature of the language.

Current development of Explora includes new physicochemical descriptors, additional ring types (e.g. bridge head) and atom hybridization types.



Easy and dynamic extension of the Explora language. The extension can occur at runtime allowing the developer to shape the language according to the application context.

Core Explora: Atom context Symbols, charge, type, requiredRingSize, ...

Extended Explora: Atom context Symbols, charge, type, requiredRingSize, ...

Dynamic Explora extension: addSymbol("index", IndexDescriptor.class)

Example: type == C and index == 1

## Conclusion and perspectives

Explora's key idea is to enrich structural queries with constraints that can be expressed in a domain specific language. We designed Explora to capture these constraints in the form of scripts that are easy to interpret and fast to execute. Explora's extensible library of functions combined with logical operators provides a new structural query paradigm. The core language can be easily and dynamically extended with new functionality written in JAVA. In the future Explora will evolve into a fully featured chemical language supporting variables, flow control statements, libraries, parallel processing, etc. We hope that Explora will become a powerful, natural, intuitive and attractive way to build chemical information workflows without requiring programming skills.

1. Derek Nexus, [https://www.lhasalimited.org/derek\\_nexus/DX/](https://www.lhasalimited.org/derek_nexus/DX/)  
 2. Meteor, [https://www.lhasalimited.org/meteor/general\\_information/](https://www.lhasalimited.org/meteor/general_information/)  
 3. SYBYL Line Notation (SLN), A Versatile Language for Chemical Structure Representation, Sheila Ash, Malcolm A. Cline, R. Webster Homer, Tad Hurst, and Gregory B. Smith, J. Chem. Inf. Comput. Sci., 1997, 37 (1), pp 71-79  
 4. SMARTS, <http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html>  
 5. Molecular Query Language (MQL), A Context-Free Grammar for Substructure Matching, Ewgenij Proschak, Jörg K. Wegner, Andreas Schüller, Gisbert Schneider, and Uli Fechner J. Chem. Inf. Model., 2007, 47 (2), pp 295-301  
 6. L-Patterns, A novel perspective on structure class definition and search in chemical structural spaces, Hanser Th.; Rosser S.; Werner S., 5th Joint Sheffield Conference on Chemoinformatics, Sheffield, UK 2010  
 7. The Definitive ANTLR Reference, Terence Parr Pragmatic Bookshelf, 2007 ISBN:978-0-9787392-5-6, ISBN 10:0-9787392-5-6  
 8. An Algorithm for Subgraph Isomorphism, J. R. Ullmann, 1976. An Algorithm for Subgraph Isomorphism. J. ACM 23, 1 (January 1976), 31-42.