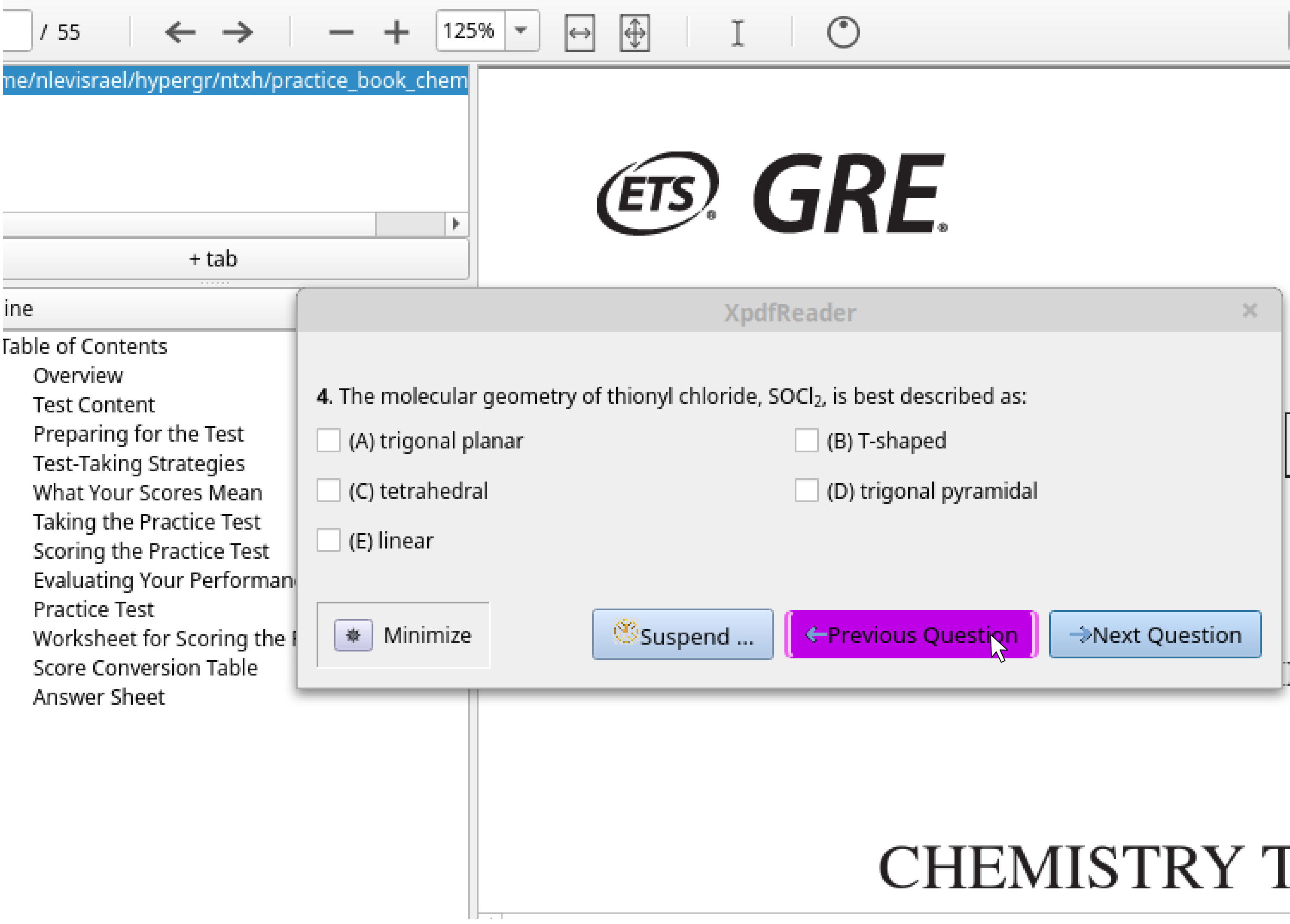
ETS Plugins

The following pages will describe a proposed ETS Plugin Framework (**EPF**). The goal of **EPF** is to integrate document viewers with test-taking technology and with scientific and multi-media applications, so that multiple applications can be unified into interactive, multi-media pedagogic platforms. With **EPF**, applications can be retrofitted to serve as teaching and testing tools. This may involve taking practice tests (or even real exams) within scientific applications, or using instructional materials co-ordinated with technical applications to help students master technical subject-matter. Moreover, while studying for exams, interactive multimedia presentations can help students benefit from sophisticated data visualization and **3D** graphics tools.

A second goal of **EPF** is to provide machine-readable structural representations of publication manuscripts, which document viewers may use to introduce additional pedagogic content: review questions, student instructions, glossaries, reading assignments, and so forth. For this technology, each publication may provide a "Semantic Document Infoset" (**API**), which divides manuscripts into textual units (sections, paragraphs, sentences, etc.) and identifies document elements such as glossary terms and figure illustration. ETS plugins can then examine a publication's **API** so as to determine how to augment the underlying document with additional instructional and/or multimedia features.

Figure 1: A Sample Practice-Test Question within XpdfReader



*Employing **EPF** as test-taking software*

One use-case for **EPF** is to embed capabilities to host or administer tests within document viewers and scientific applications. At the most basic level, these features would include separate windows which show questions and answer choices, with options to navigate between questions. When used for test preparation, students can use this format as a convenient simulation of an actual test, while also browsing back and forth between the question/answer window and the host application. As depicted in Figure 1, question/answer plugins may be embedded in a **PDF** viewer which loads a practice test in text format (in this case practice for the Chemistry **GRE**), so students have the option of reading through the practice materials as a book-like publication and also conducting a practice test session in a more exam-like fashion, viewing one question at a time.

*Writing exams with LaTeX and **XPDF** plugins.*

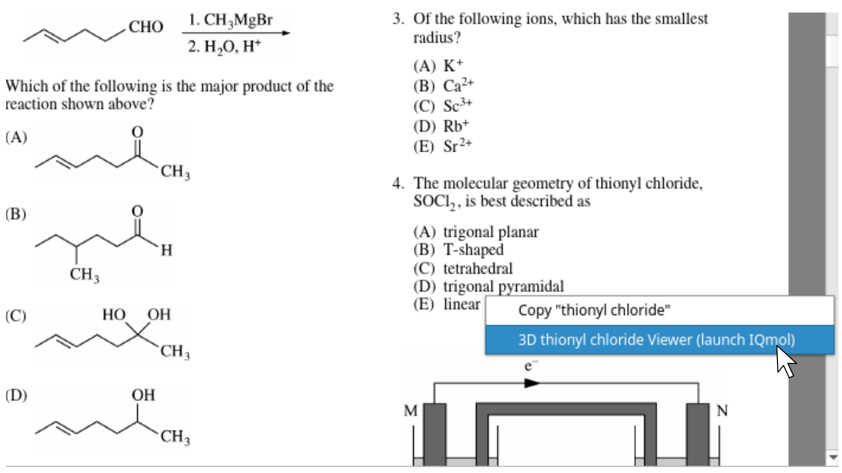
To support this technology, **EPF** distributions can include LaTeX packages for composing exams which allow the questions and answers to be automatically extracted from the TeX sources, packaged in a structured format (such as **XML**), and placed as an embedded file in the generated **PDF** document. This embedded data can then be read by **EPF** plugins to create the question/answer windows. In manuscripts (such as the ETS **GRE** practices) where questions are printed as part of the publication text, the LaTeX code can store questions' **PDF** coordinates so that the document automatically scrolls while students work their way through a practice test session. On the other hand, the same technology can be used to add review questions and answers to documents which are not expressly designed as test-prep materials, such as textbooks and research papers. In this latter case, question/answer windows may be synced to sentences or paragraphs in those publications which are relevant to the review question that the student is currently reading in a question/answer window.

Remote Test Administration

Similar technology can also potentially be used for remote administration of actual exams. Such a use-case is consistent with the current trend toward online test-taking — a trend which may accelerate in the near future as educational opportunities tick upward in remote areas and countries, where students cannot easily travel to testing centers (or even due to situations such as Covid-19 that disrupt conventional testing logistics). Remote administration via **EPF** is feasible if the plugins communicate students' answers in real time to **API**s provided by the testing service. Architecturally, the overall setup is then similar to online test-taking, except that students use technical applications or document viewers instead of web browsers for the actual test-taking.

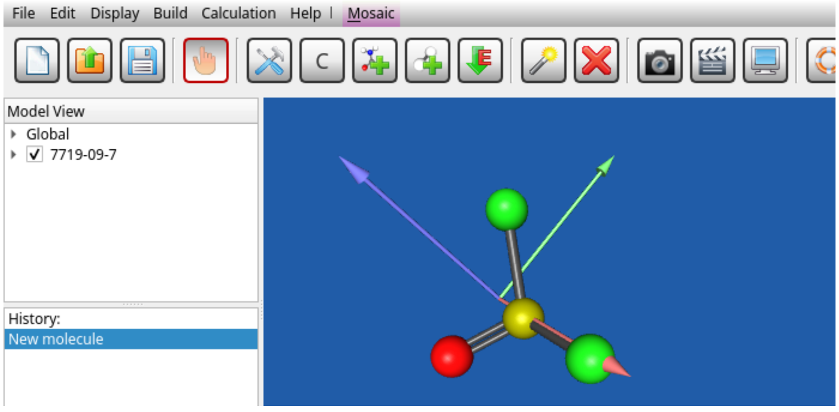
*How **EPF** enables multi-application networking*

In addition to potential test-taking features, **EPF** also supports connecting multiple applications, so that students benefit from an interactive, multi-media pedagogical environment. **EPF** refers not to a single plugin, but a toolkit for implementing ETS plugins to be embedded in many different applications. These plugins should be sufficiently similar that students or instructors familiar with an ETS plugin in one context (chemistry, for example) would quickly understand how to use plugins present in a different context. An important **EPF** feature is that distinct ETS plugins would be able to communicate with each other. In particular, plugins for document viewers would send data to plugins for scientific or multimedia applications, so that students could access multimedia content linked to manuscripts, such as test-preparation materials, that they are currently reading.



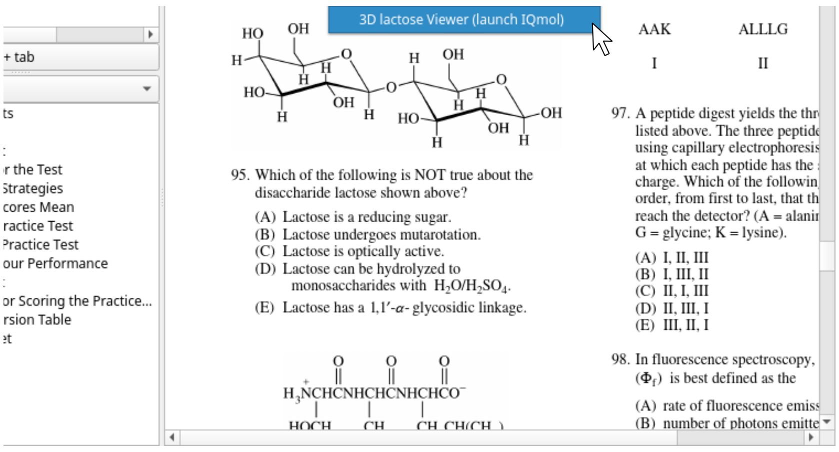
For a concrete example of advanced functionality that can be achieved by connecting two distinct **EPF** plugins, consider a student studying the **GRE** Chemistry Practice Book published by ETS. This book has sample questions such as (number 4, page 11) **The molecular geometry of thionyl chloride, $SOCl_2$, is best described as** *(A) trigonal planar, (B) T-shaped, (C) tetrahedral, (D) trigonal pyramidal, or (E) linear*. To understand this question/answer, it may help students to view a **3D** model of thionyl chloride, which can be done through molecular visualization software such as IQmol. Accordingly, this specific question in the book may be associated with Molecular Data file for $SOCl_2$ (this file is available from the "Chemical Abstracts Service" database). The relation between

Figure 2: Thionyl Chloride in IQmol



the specific textual location (where the practice Question 4 is presented) and the supplemental Molecular Data file would be asserted in the Semantic Document Infoset, and read by a document viewer (e.g., **XPDF**). The **XPDF** plugin would then launch IQmol and send the molecular file to the IQmol ETS Plugin, with instructions to load this file into an IQmol session (see Figure 2). The end result would be that the student, with a single click (such as selecting a visualization action from a context menu on the practice question) has access to an interactive **3D** graphic representing thionyl chloride. (Of course, analogous functionality would be available for any chemical compound with multimedia files in such formats as Molecular Data, Protein Data Bank, or Chemical Markup Language).

The data sent between **EPF** applications may be more complex than a request to open a single multimedia file. Suppose a student reading the GRE Chemistry practice exam launches IQmol a second time — perhaps in conjunction with a later question (95) about the molecular structure of glucose. In this case, the plugin can send information not only about the present request but about the student's prior usage; in particular the fact that he or she had previously viewed the $SOCl_2$ file. The **EPF** plugin on the IQmol side can then load the prior file along with the new one, so the student can browse back to prior application-states if desired (see the Model View panel on Figure 3).



In general, the functionality provided by each ETS plugin will depend in part on the host application where the plugin is embedded. An IQmol plugin would load cheminformatic files and may activate IQmol's analytic capabilities in the domain of chemistry, whereas a plugin in Data Visualization applications (such as ParaView) could open quantitative data sets with 2D or 3D views (via surfaces, scatter-plots, bar charts, etc.) and activate statistical calculations. Certain functionality, however, would be shared among all ETS plugin, including a dialog window to show basic plugin information (see figure at left) and also a more

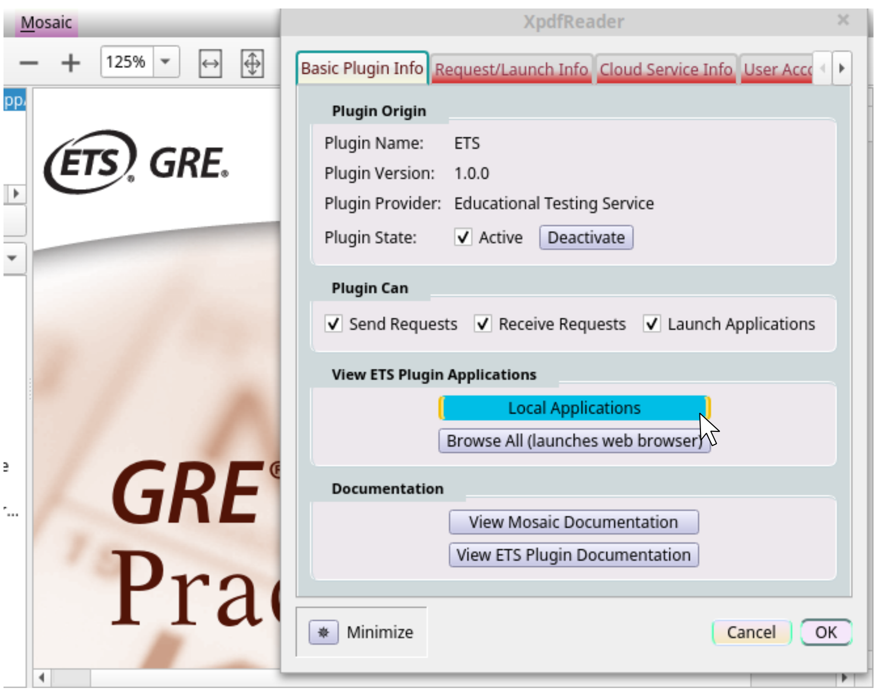
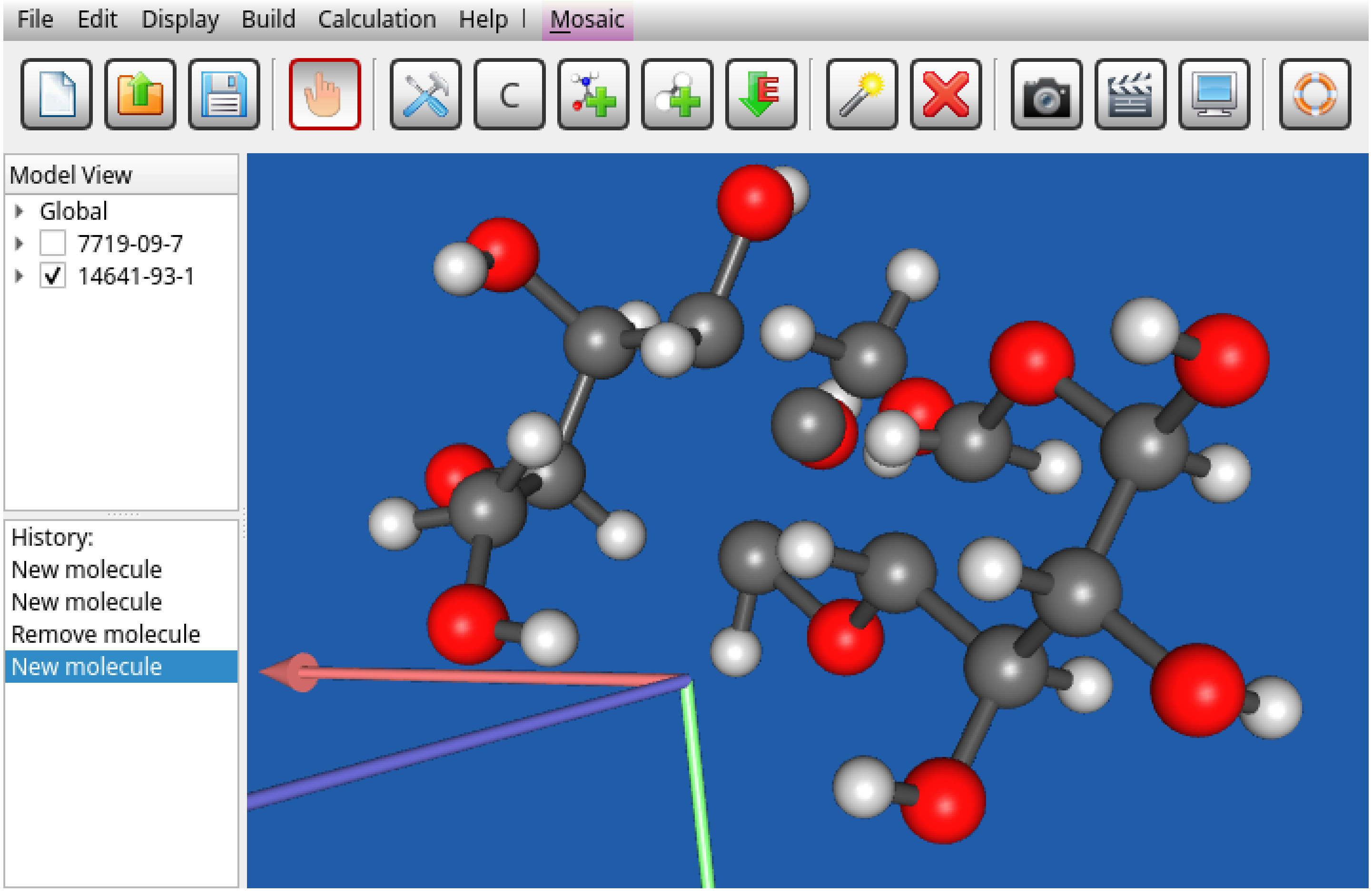


Figure 3: Lactose in IQmol



detailed review of data transmitted between applications via plugins. The "request info" tab allows students, instructors, and plugin developers to see information about the request which caused the current application to be launched or to open a specific file (see Figure 4).

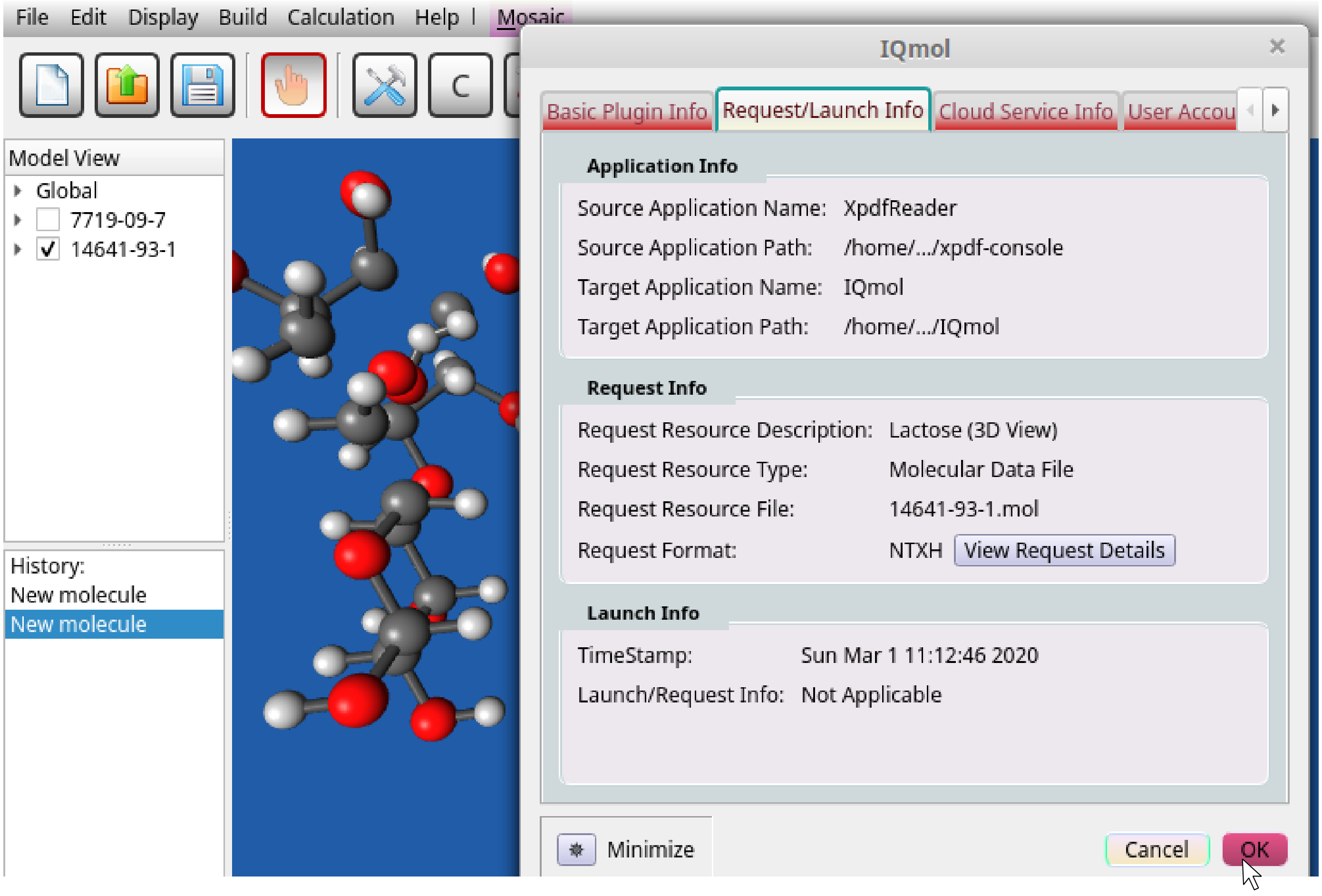
In addition to data visualization, scientific applications can help students understand concepts which are covered by a test. For example, a later **GRE** Chemistry practice question concerns Orbital Angular Momentum. To understand this topic, students may benefit from hands-on experience calculating and visualizing Molecular Orbitals in IQmol. In this scenario, once again, the practice book may be linked to IQmol through the Orbital Angular Momentum question. However, in this case, instead of showing a single molecule, IQmol could load an interactive tutorial — provided by the ETS Plugin — explaining the Canonical Orbital Surfaces features in IQmol and enabling students to explore these with a variety of different molecules.

Use-Cases for the Proposed ETS Plugin Framework

The above figures illustrate some of the canonical use-cases for ETS plugins: students read textbooks, articles, or test-preparation materials which are be enhanced with multimedia content. For other use cases, consider the following scenarios:

Scenario 1: Using scientific application as pedagogical tools Teachers often instruct students to download and install software relevant to the course curriculum, and this software can potentially be an essential part of the course content. Instructors may (1) use the visualization capabilities of these domain-specific applications to help students understand the concepts covered in class; (2) provide instruction in how to use the software as part of the curriculum; (3) evaluate students' understanding of the software as part of their assessment of students' mastery of the curriculum; and (4) use applications' analytic features as an overview of analytic or quantitative methodologies relevant to the course's subject matter. In the case of IQmol, features such as energy minimization, plotting orbitals, calculating vibrational frequencies, and many other chemphysical computations

Figure 4: Request Information in IQmol



provide an overview of scientific concepts which might be covered in a Chemistry class.

Science applications as teaching tools

To facilitate the use of scientific applications as teaching tools, **EPF** plugins help instructors personalize the applications which their students use in conjunction with course curricula. Again using IQmol as a case-study, an ETS plugin could show students a list of molecular examples discussed in class (based on data provided by the instructor) and allow students to view the corresponding **3D** molecular graphics accordingly.

Scenario 2: Authors or instructors analyze publications' content to develop teaching materials When documents are read in an educational context, they become part of a larger ecosystem that extends beyond publication itself — an ecosystem of assessment, test preparation, curricular design, and overall teaching environments (labs, classrooms, and so forth). Pedagogically using publications can involve extracting information from texts, or laying on additional content: inserting review questions at strategic points in the text; compiling glossaries indicating when technical terms are first used; foregrounding graphics and figure illustrations and encouraging students to explore them in depth (for instance, performing for themselves the calculations which are plotted in a statistical diagram; or identifying the axes, scales of measurement, dimensions, free and dependent parameters, and other mathematical details of a quantitative illustration). Both of these kinds of operations — extracting or adding content — require computational access to a Document Infoset providing structured access to publication content.

Computer code which operates on manuscripts composed in coordination with **EPF** can therefore act in consort with **EPF** plugins. For example, review questions or glossary definitions may linked to specific points in the text; this supplemental content may then be packaged along with the original publication. With suitable **EPF** plugins, the document viewer can then process the review material and superimpose pedagogical questions, comments, or instructions on the underlying document text.

Scenario 3: Authors Develop Educational Materials Targeting Specific Applications In this scenario, authors compose books, articles, or test-preparation materials with the anticipation that readers will use specific software applications to enhance their reading experience. This sort of interrelationship between publications and external software is presupposed at a rudimentary level as soon as authors link documents to specialized multimedia files. For example, files in the ParaView Data (PVD) format are intended to be used by the ParaView software; as a result, documents which reference such files presuppose that readers will have ParaView installed on their computer, for the full reading experience. Similarly, files in cheminformatic formats such as Protein Data Bank (.pdb) or Chemical Markup Language (.cml) need to be opened with chemistry-related software like IQmol.

Although it is not a prerequisite for EPF plugins, one tool which can help authors customize their audience's reading experience is the Hypergraph Text Encoding Protocol (HTXN), discussed in greater detail below.

The HTXN Protocol

HTXN defines a protocol for (1) encoding manuscripts' character data and (2) defining document structures via graphs whose nodes correspond to ranges in a character stream. HTXN therefore uses "standoff annotation" (i.e., character encoding and document structure are defined in isolation from one another), and can be employed to encode manuscripts in many different markup formats. EPF and HTXN are autonomous technologies (each may be used apart from the other), but they form a natural pairing, encompassing both the character/orthographic data and the document structure of manuscripts for publication.

HTXN and multimedia content

The central goal of HTXN is to support the new generation of publishing technologies, where conventional document formats are increasingly being supplanted by digital, multimedia reader experiences. The traditional manuscript (the "primary" resource which is cited and downloaded) is, accordingly, often networked with a package of supplemental (or "secondary") resources. HTXN is designed to rigorously document these multimedia networks, enabling e-readers and domain-specific applications to be integrated so that users may easily access multimedia content.

The generic term "multimedia content" actually encompasses multiple phenomena:

Multimedia Files Individual files representing audio, video, or 3D graphics content. These files may be linked from specific locations in the primary manuscript, or even embedded within manuscripts when they are published in PDF format.

Data Sets and Data Visualization Publishers increasingly emphasize sharing research data alongside texts, so readers can verify or even attempt to replicate claimed results. Data sets are also a form of multimedia content because, apart from being aggregates of raw data, data sets are almost always viewable via interactive, visual content: charts, diagrams, or plots to visualize the information holistically, or interactive tools to examine or navigate through the data set at finer scales.

Application Networks Another genre of multimedia content involves resources which may only be experienced through specialized software. This classification encompasses content from particular scientific or technical domains, which is encoded in domain-specific formats: representations of molecular structures, archaeological sites, image-processing data, wave-forms for signal processing, sentence-parses for linguistic analysis, and so forth. To conveniently access this kind of multimedia content, readers need to use software which can send signals to the specialized applications having the capability to recognize the domain-specific formats and translate them to interactive, visual presentations. In short, publication viewers (e.g., e-readers) need to participate in multi-application networks, where data can be sent and received between each component. One way to achieve this is via protocols, such as **EPF**, shared between both e-readers and target applications.

Publications-as-Applications In some cases, publications themselves are a form of multi-layered multimedia content. This applies to publications which are not simply read from start to end, but instead naturally lend themselves to a reading process which navigates back and forth between different sections of the text, or juxtaposes different sections to be visible at the same time. Testing materials and test preparation are a canonical example of such layered reading, where exam questions, instructions, supplemental materials (such as passages for reading-comprehension assessment), and comments or analyses about answers (in the case of prep materials) embody different layers which students may wish to view side-by-side. In these cases, e-readers cannot simply treat the publication as a single file. Instead, the manuscript needs to identify text segments which can be factored into different layers, and the e-reader needs to implement text-viewers which allow each layer to be viewed in separate windows, with readers able to juxtapose and position the windows as desired.

HTXN represents publication manuscripts using structures which rigorously document publications' multimedia content and multi-application networking requirements. This detailed multi-media support has several dimensions:

1. Defining points in the manuscript where multimedia files are linked or embedded: this involves annotating locations in the manuscript with hyper-references to multimedia files (audio, video, etc.) which readers should be able to access when they reach the corresponding point in the text.

HTXN cross-references to data-sets

2. Establishing granular cross-references between publications and multimedia content: this is a more complex case where manuscript locations have to link *to* or *from* limited *portions* of the corresponding multimedia resources. For example, a passage in the manuscript may discuss a single sample within a data set; or may explicate a particular facet of the data set, such as an individual column in a tabular information space, or a specific set of statistical parameters against which quantitative operations are performed. These scenarios call for bi-directional cross-references between the data set and the publication, wherein the granular data-set facet topically relevant to the corresponding manuscript location (the sample, table-column, parameters, etc.) is formally isolated and declared as a reference-target.

HTXN cross-references to multi-media assets

3. Cross-references may also be defined between publications' non-textual or non-paragraph content and corresponding multimedia resources. For example, tables or diagrams visually presented in a manuscript may be liked to statistical data from which the figures are derived. A similar situation applies when visuals included in a publication are linked to multimedia resources which represent the same information in a different experiential register: a **PDF** document may include a two-dimensional graphic which is created by taking a camera shot of a **3D** model, which readers may also experience with a **3D** graphics engine; or a publication may reproduce a graph or scatter-plot derived from a data set, where data visualization software can represent the same information in a more interactive medium, with parameters plotted as curves or surfaces in a **3D** ambient space, or where systems are visualized as systems evolving over time.

A summary of the **HTXN** protocol, sample documents which compile to **HTXN**-encoded text, and more information about plugins, are all available on request from Linguistic Technology Systems.