An ETS Plugin Framework for Enhanced Test Preparation

We propose an ETS Plugin Framework (**ETSPF**) whose goal is to augment document-viewing applications (for example, **PDF** viewers), along with scientific and multi-media applications, with instructional and test-preparation features. ETS plugins would allow document viewers to launch and share data with a diverse array of applications (including software specialized for scientific disciplines such as chemistry, physics, biology, and medicine), supporting a diverse array of data types and file formats. Document viewers would therefore support an interactive, multimedia reading experience to an unprecedented degree, which would especially benefit students during test preparation. In particular, students would have at their disposal pedagogically stimulating multimedia presentations that offer sophisticated data visualization and **3D** graphics tools, customized for individual subjects (**3D** molecular models for chemistry, **3D** tissue models in for biology, and so forth). ETS plugins could also enhance document viewers with instructional features introduced as supplements to the documents which students are reading, such as review questions or assignment instructions.

ETSPF for Scientific and Technical Applications

ETSPF 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 **ETSPF** 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 documents (e.g., test-preparation materials) that they are currently reading.

How ETSpf enables multi-application networking

For a concrete example of advanced functionality that can be achieved by connecting two distinct **ETSPF** plugins, consider a student reading through the ETS **GRE** Chemistry practice test. 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 with the aid of 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 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

Figure 1: A Thionyl Chloride Question in XpdfReader

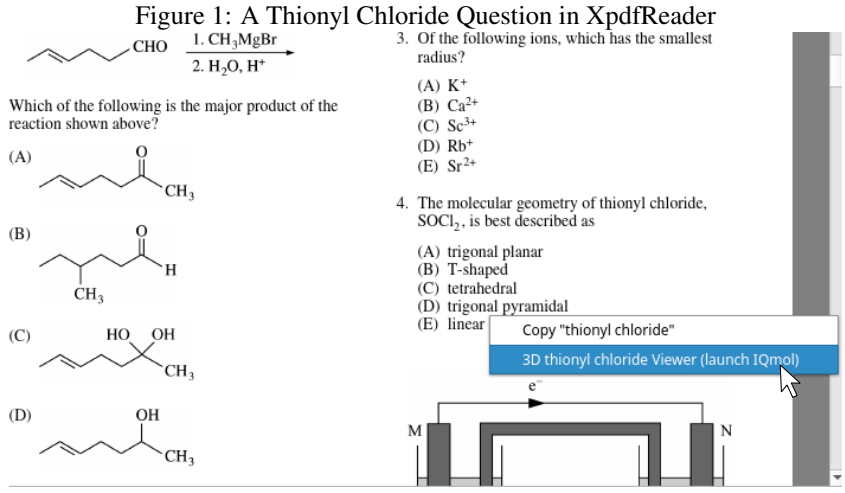
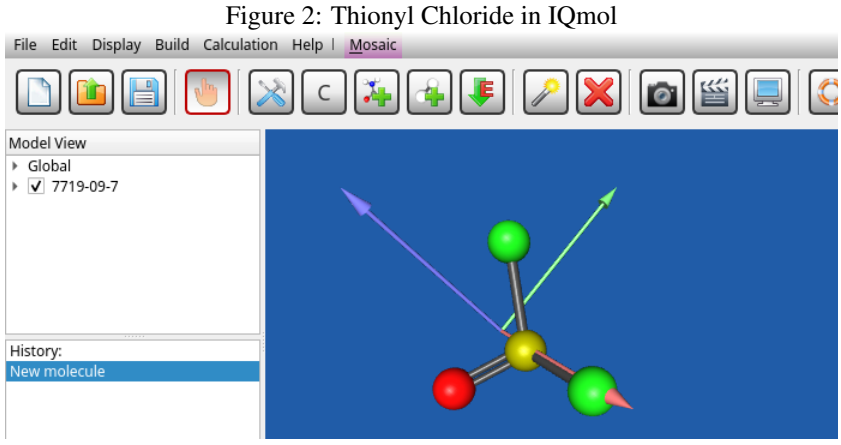


Figure 2: Thionyl Chloride in IQmol



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).

ETSpf features for keeping track of students' previous activity.

The data sent between **ETSPF** applications may be more complex than a request to open a single multimedia file. Suppose a student reading through the GRE Chemistry practice exam launches IQmol a second time — perhaps in conjunction with a later question (number 95 in the test — see figure at right) about the molecular structure of lactose. 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 **ETSPF** 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 4).

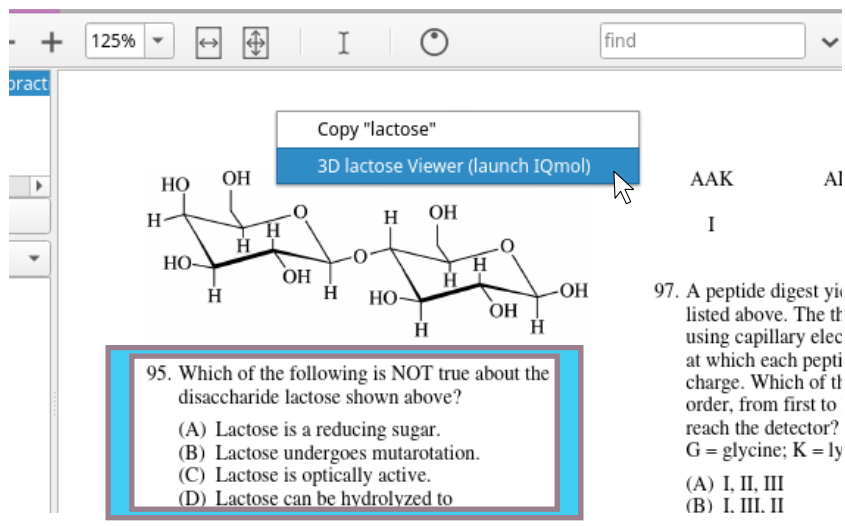


Figure 3: Plugin Info in XpdfReader

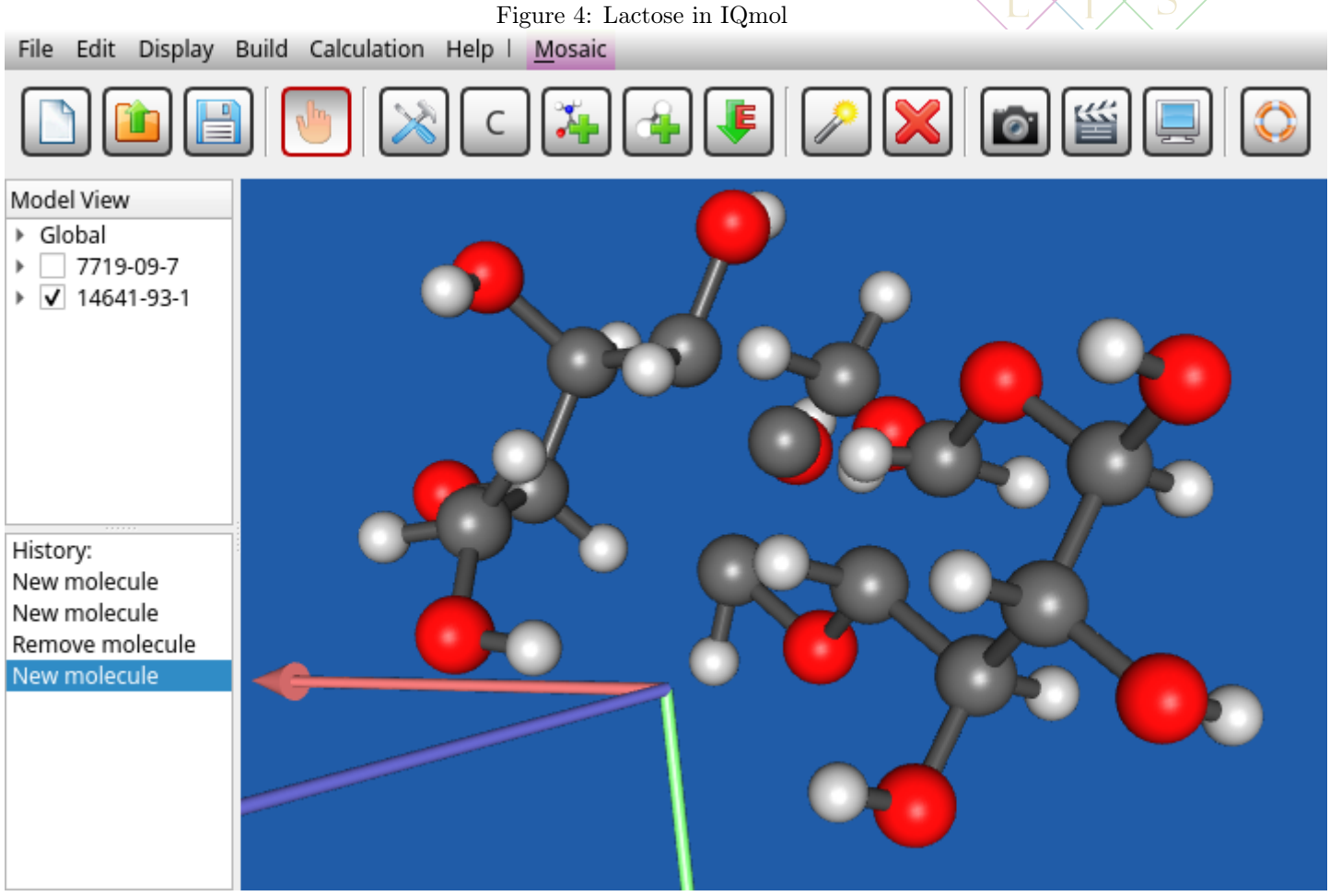
In general, the functionality provided by each ETS plugin would depend in part on the domain of the host application in which the plugin is embedded. For example, an IQmol plugin would load cheminformatic files and may activate IQmol's analytic capabilities in the domain of chemistry, whereas a plugin for applications in the domain of quantitative/statistical analysis and data visualization (such as ParaView) would open quantitative data sets with 2D or 3D views (via surfaces, scatter-plots, bar charts, etc.) and perhaps activate statistical calculations. Nevertheless, certain functionality would be shared among all ETS plugins, which would include a dialog window to show basic plugin information (see Figure 3) as well as a more detailed review of data transmitted between applications via plugins. Specifically, the "request info" tab would allow students, instructors, and plugin developers to see information about the request which prompted the current application to be launched and/or to open a specific file (see Figure 5).

ETSpf Tools for Composing Test-Preparation Materials

ETSpf data in embedded files

In most cases, **ETSPF** plugins for document viewers such as **XPDF** would draw information from **PDF** files (or files in other formats, e.g. **EPUB** or **HTML**) to implement pedagogic enhancements (such as integration with scientific and multimedia applications). This **ETSPF**-specific data can be placed in a separate file embedded in **PDF** or **EPUB** documents (or inserted as non-display contents in **HTML**). When a document is opened, the **ETSPF** plugin would then extract the embedded file so as to read **ETSPF**-specific data about the document — in particular, to identify **PDF** coordinates for document elements requiring special **ETSPF** actions. For questions (4) and (95) as illustrated above, the relevant **ETSPF** action would be an option to view the question-specific molecular files in IQmol. **ETSPF** data

Figure 4: Lactose in IQmol

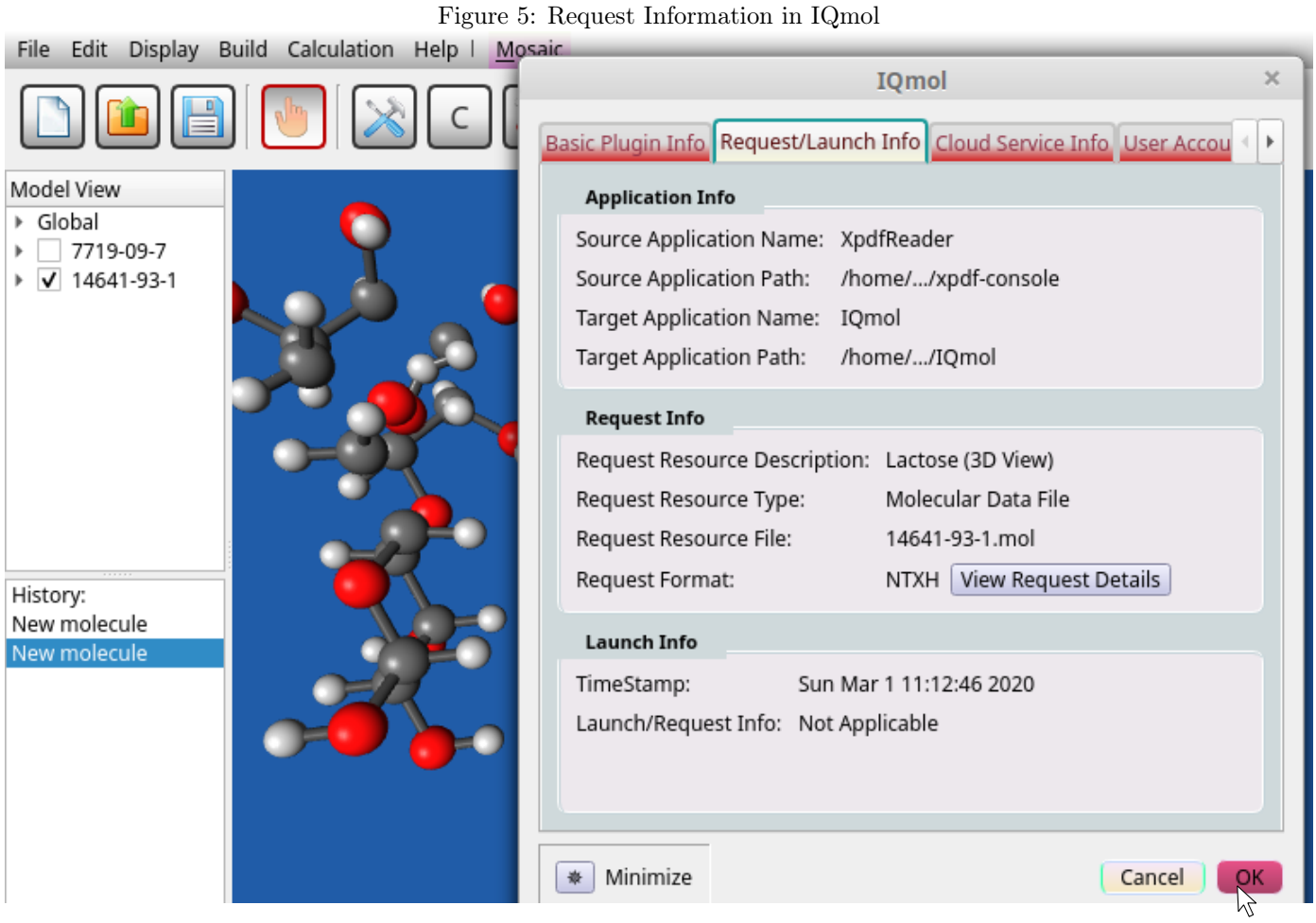


is needed in order to map the textual boundaries of the question (and its multiple-choice answers) to on-screen coordinates, so that context menus can be customized for each question.

Semantic Document Infosets (SDIs)

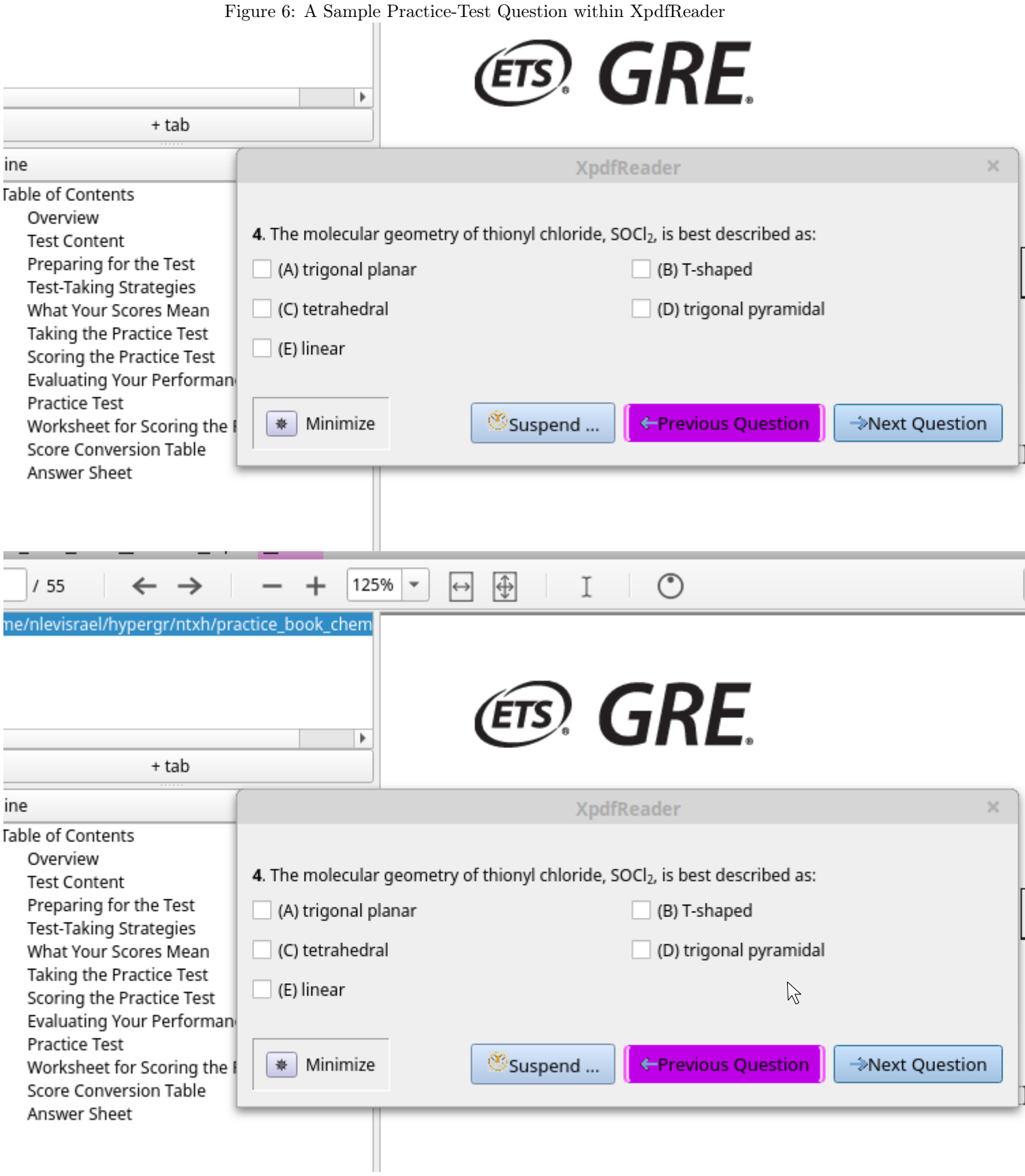
To support these capabilities, **ETSPF** would include tools to help compose publications (such as test-preparation materials) that embed what we call a "Semantic Document Infoset" (**SDI**), which divides manuscripts into textual units (sections, paragraphs, sentences, etc.) and identifies document elements such as technical terms (which may be compiled into a glossary) and figure illustrations. **ETSPF** code can then examine a publication's **SDI** to generate machine-readable structural representations of publication manuscripts, which document viewers may use to augment the underlying document with additional instructional and/or multimedia features — review questions, student instructions, glossaries, reading assignments, and so forth. The **SDI** can be used to guide **ETSPF** plugins

Figure 5: Request Information in IQmol



when sharing data between applications — in Figure 1, for example, selecting the Molecular Data file to send to IQmol based on the screen coordinates of the context menu — but also to enhance the presentation of content within the host application. For example, Figure 6 shows how an ETSPF plugin could provide an alternative interface for viewing practice test questions, where readers can consider each question in turn, isolated in its own window.

Figure 6: A Sample Practice-Test Question within XpdfReader



Using LaTeX to generate SDI info-sets

ETSPF implementations can include LaTeX packages which automate the creation of SDI data (placed as an embedded file in the generated PDF document). This embedded data can then be read by ETSPF plugins to compose multi-application networking requests, populate question/answer windows, or introduce other kinds of pedagogic content (e.g., review questions, glossaries, or class discussion suggestions). In documents where questions are printed as part of the publication text (for example, the ETS GRE practices), the LaTeX code can store questions' PDF coordinates so that the document automatically scrolls while students work their way through a practice test session. Alternatively, the same techniques 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.

HTXN (Hypergraph Text Encoding Protocol) Specifications

As an additional feature, **ETSPF** plugins would implement a protocol which we call **HTXN** (for "Hypergraph Text Encoding"). The 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 **HTXN** protocol 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 different markup formats (both LaTeX and **XML**, for instance). In the context of **ETSPF**, **HTXN** would be used to encode document information and text within the **SDI**.

LTS can provide a more detailed overview of **ETSPF** with additional use-cases, technical information about plugin code, and sample **HTXN**-encoded documents, on request.