Hypergraph Text Encoding Protocol (HTXN)

This paper will summarize the Hypergraph Text Encoding Protocol (HTXN) and discuss its applications for testing, education, and the development of test and test-preparation materials.

HTXN is a new format and protocol for representing publications. The central goal of HTXN is to support a new generation of publishing technologies, where conventional document formats are increasingly being supplanted by digital, multi-media reader experiences. In the contemporary publishing paradigm, individual publications are often linked with other forms of digital content: multi-media resources, research data sets, machine-readable representations of document text, and domain-specific sofware applications (used to study or visualize the case-studies or research findings discussed in publications). The conventional manuscript (the "primary" resource which is cited and downloaded) is then networked with a package of supplemental (or "secondary") resources. The HTXN protocol is designed to rigorously document these multi-media networks, enabling e-readers and domain-specific applications to be integrated so that readers may easily access and experience multi-media content.

HTXN for Multi-Media

The generic term "multi-media 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 accompanied by 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 multi-media 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 multi-media, 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. Publishers can provide this functionality to readers by implementing special e-readers and, in addition, writing plugins (or collaborating with external application developers) to ensure that applications networked with e-readers are properly aligned with the e-readers themselves.

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. A canonical example of such layered reading is testing materials and test preparation, 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), each form different layers which students may wish to view side-by-side. In these cases, e-readers cannot simply treat the publication as one single ePub or PDF 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' multi-media content and multi-application networking requirements. This detailed multi-media support has several dimensions:

1. Defining points in the manuscript where multi-media files are linked or embedded: this involves annotating locations in the manuscript with hyper-references to multi-media files (audio, video, etc.) which readers should be able to access when they reach the corresponding point in the text.
2. Establishing granular cross-references between publications and multi-media content: this is a more complex case where manuscript locations have to link *to* or *from* limited *portions* of the corresponding multi-media 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.
3. Cross-referances may also be defined between publications' non-textual or non-paragraph content and corresponding multi-media 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 inluded in a publication are linked to multi-media 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.

HTXN for Teaching Materials

ETS Plugin Framework The proposed ETS Plugin Framework would create a common code base that can be used to implement ETS-specific plugins to applications spanning a range of academic disciplines and subject areas. The primary goal of these plugins would be to connect e-reader software — applications for viewing test-preparation and course materials — with domain-specific software which instructors may use as teaching aids. With respect to GRE exams in fields such as biology, chemistry, or physics, domain-specific applications might include bioinformatics, molecular visualization, or physical simulation tools, respectively. ETS plugins would help scientific applications become more valuable as classroom tools.

In particular, with ETS plugins these applications can (1) receive signals from document viewers (such as PDF viewers) to automatically display multi-media content and (2) display course and instructional materials. As a concrete example, consider molecularal visualization software such as IQmol. Via an ETS plugin, IQmol could (1) render a **3D** image given data, in a chemical file format, received from an e-reader with a similar plugin; and (2) display instructions, questions, assignments, or course-related content (such as graphics for chemical compounds discussed in class) provided by instructors.

Cloud Support for ETS Plugins Supplementing the functionality described in the preceding paragraph, Cloud Services can be used to connect individual applications to content and curricula specific to individual courses. Because most scientific applications are implemented as desktop software, the hosting of the cloud back-ends to support these features would be a natural fit for LTS's "Native Cloud/Native" (NCN) protocol, which is specifically designed to integrate desktop front-ends with Cloud/Native services. According to this architecture, NCN services would host information specific to individual courses, students, and instructors. When a student launches an application with an ETS plugin, the plugin would retrieve data pertaining to that student and to his or her classes. As appropriate, the plugin could then instruct the host application to load specific content, and/or present questions or instructions supplied by the instructor. For example, IQmol could load a list of molecular files corresponding to chemicals studied in the students' class. The ETS Plugin Cloud back-end would also be used by document (e.g., PDF) viewers to obtain information needed to properly route signals to other applications using ETS Plugins.