Hypergraph Text Encoding Protocol (HTXN)

The Hypergraph Text Encoding Protocol (**HTXN**) is a new publication format which can generate documents in a variety of different representations (such as LaTeX, **XML**, **HTML**, and **RDF**). **HTXN** enables publication workflows to combine two or more existing formats (e.g., LaTeX and **XML**). A document with **HTXN** encoding will generate LaTeX or **XML** "views" which can each be constructed for use at different stages in the document-preparation pipeline. **HTXN** is also designed to work with publications that are embedded in a larger data or multi-media ecosystem — in particular, documents that are paired with data sets or multi-media content that need specialized software.

Benefits

Mix-and-Match Formats **HTXN** employs "stand-off" annotation, which allows multiple markup protocols to be defined simultaneously on the same document. In addition, **HTXN** uses a flexible character-encoding protocol that ensures compatibility with diverse text representations (such as **XML**, LaTeX, Unicode, and **Qt**/QString). Via secondary documents or "views" (generated from the primary **HTXN** document), **HTXN** can be used wherever **XML** or LaTeX (or other formats) are desired.

Fine-Grained Character Encoding The **HTXN** protocol includes more detailed document structure information compared to conventional markup. For example, **HTXN** identifies sentence boundaries and punctuation features, disambiguating logically distinct but often visually identical characters (such as dots/periods or dashes/hyphens, which may have different structural meanings in different contexts).

Multi-Media and Data Set Integration **HTXN** is optimized for sophisticated publishing environments which combine conventional natural-language texts with interactive data sets and software. Contemporary publishing increasingly sees academic texts as part of a multi-faceted ecosystem, where publications, research data, and interactive software can be packaged into multi-media presentations. **HTXN** ensures that documents can be properly experienced within these augmented publishing platforms.

Complex Anchors and Interoperability

HTXN introduces a variation on micro-citations and conventional **HTML** or LaTeX anchors/labels, which we call "complex anchors", to facilitate interoperability with research data sets and external software. Complex anchors permit concepts, terms, and data structures to be tracked and cross-referenced between publications and their associated data sets or supplemental software. Consider a table-structured data set whose columns represent scientific concepts or statistical parameters that are discussed in an associated publication (e.g. article or book chapter), or where data from a specific table is presented graphically as a figure within the publication. Via complex anchors, these kinds of thematic relationships between data-set elements and corresponding points in the research text may be made explicit, so that data-set applications (including software expressly implemented for a newly published data set) can recognize references "pointing in" to the publication. In the context of tabular data, data-set software could provide context-menu actions associated with each column (or, in the case of figure illustrations, the overall table), linking to an anchor in the text where the relevant concepts, terminology, or visualizations are explained.

This level of interoperation can be achieved by implementing document viewers (such as a **PDF**

viewer) embedded in scientific software or software expressly designed for a data set. With embedded viewers, the operation of opening associated publications to specific anchor-points (such as figure illustrations) can be integrated as an interactive extension to their host applications, triggered by menu items or other GUI components. To support this interoperability, HTXN provides anchors whose identifiers and embedded data can be customized to work with domain-specific host applications.

Complex Exolinks and Interoperability

HTXN "exolinks" refer to data or multi-media content *outside* the text (so they are effectively the converse of complex anchors, which support references pointing *inside* the text from external software components). Complex exolinks designate data or multi-media content — such as video, audio, or 3D (scenes or panoramic-photography) graphics — which are outside the text, but potentially part of a larger publication package wherein an article is accompanied by data sets and/or multimedia content. Traditional external/HTTP references in HTML or PDF are designed to reference resources on the World Wide Web, but not content downloaded in a package along with publications themselves. Therefore, special reference formats are required for document viewers to properly handle links in the text that point to non-web content (which is a technical limitation present in conventional document encoding formats). HTXN will ensure that, when accessing multi-media content, properly encoded signals will be sent between the document-viewer components and applications which support the relevant multimedia file types (potentially host applications where the document viewers themselves may be embedded).

As a concrete example, consider a chemistry paper which is paired with 3D descriptions of a particular molecule. In order for readers to view this content, a signal needs to be sent to compatible molecular-visualization software (such as IQmol — to cite a Qt-based, open-source case-study). Analogously, when reading a paper concerning 3D radiology, signals may be sent to — or be read on components embedded within — 3D viewers such as MeshLab (again Qt/open-source). To support interoperability requirements as shown by these examples, the HTXN specification includes an HTXN-specific "Native Application Network" (HNaN) protocol to connect document viewers with external and/or host applications; HTXN exolinks encode data which ensures that the concordant HNaN signals are properly generated.

HTXN has numerous features for interoperation with software applications where document viewers may be embedded, or which support HNaN via plugins. In particular, HTXN parsers and generators are provided via a Qt/C++ library that can be dropped in to the source code of native C/C++ applications, so it is straightforward to add HNaN to existing software. Based on Qt (a C++ GUI and application-development framework), HTXN can leverage the full range of Qt networking or front-end features (for instance, access to publishers' APIs can be introduced as a processing stage during document preparation). In addition, HTXN's graph-based parsing framework and hypergraph-based annotation model can be customized and paired with project-specific GUI components for viewing and accessing data sets. Via such customizations, HTXN can be adapted to provide a domain-specific publication platform for specific research projects or environments (individual journals, book series, research labs and academic departments, etc.). HTXN's "reference implementation" includes a specially-designed "Next-Generation" markup language (NGML), as one way to build HTXN files. The NGML library is transparent and distributed in source-code form, so it can be modified to create domain-specific markup styles. A customized version of NGML can thereby be used to construct HTXN documents according to each individual lab's or journal's specifications.