NA3 HTXN Plugin Framework

This paper will describe the plugin mechanism for **HTXN** (the Hypergraph Text Encoding Protocol), which is part of the **NA3** Native Application Framework. **HTXN** is a new document format with several breakthrough features: a LaTeX-compatible Document Object Model; the option of generating versions of the document in multiple formats, such as LaTeX and **XML**; and fine-grained representation of structural features such as sentence boundaries. When used in **NA3** applications, **HTXN** allows authors to compose manuscripts which depict or discuss data maintained within the host application. One feature of **HTXN** is the ability to call host-application procedures to generate content which is inserted or linked into a manuscript. For instance, if an author wants to insert a graphic before a specific paragraph, an **HTXN** instruction would invoke host procedures to create the figure as an external asset file, then insert code in the manuscript to include the figure at the desired point in the text.

HTXN plugins can also be used with software not explicitly engineered via **NA3**. In principle, any application popular with scientists or researchers can host **HTXN** plugins, which become valuable when researchers transition from obtaining and analyzing data to publishing their research findings. Many scientific and technical applications have a built-in plugin or extension mechanism wherein users download and install plugins which add new features; others are open-source projects where plugins can be implemented as new libraries added to the source code. **HTXN** can, in particular, be implemented via *publisher-specific* plugins, which generate documents matching publishers' internal specifications, such as **XML** Document Type Declarations (**DTD**s). This capability is beneficial to publishers because it eliminates one step in the document-preparation workflow — namely, converting a document submitted in formats such as LaTeX to in-house **XML** or **SGML** files.

Aside from practical document-preparation benefits, publisher-specific plugins may also serve as a promotional vehicle for publishers. Application users would see the options to load different publishers' plugins; the plugin documentation, as well as a splash-screen when the plugin is loaded, could describe advanced features of the publishers' platform. Publishers are branching out away from conventional text documents to incorporate data sets and multi-media content; it behooves them therefore to find opportunities to describe and promote the more advanced features of their platform. Moreover, plugins demonstrate a level of technical sophistication; implementing plugins to scientific and technical applications demands advanced software-engineering techniques, so a publisher-specific plugin signals to the academic community that the publisher is comfortable engineering or designing complex scientific-computing components. By embedding plugins in applications spanning a range of academic disciplines, publishers have a convenient promotional hook into different scientific communities.