Tools for neuroinformatic data exchange: An XML application for neuronal morphology data

Weihong Qi[1], and Sharon Crook[2]

[1] Department of Computer Science, University of Maine, Orono, ME 04473

[2] Department of Mathematics and Statistics, University of Maine, Orono, ME 04473

Abstract

We describe an Extensible Markup Language (XML) application for the storage of neuronal morphology data. The application, MorphML, is consistent with current neuronal digitization methods and is also compatible with existing simulation software environments for creating model neurons based on morphology data. We also describe a tool, nemo2xml, which generates MorphML documents corresponding to digitized neuronal morphology files for one particular data format, the Nevin or Binary Branch Tree (BBT) format.

Tools for neuroinformatic data exchange: An XML application for neuronal morphology data

Weihong Qi[1], and Sharon Crook[2]

[1] Department of Computer Science, University of Maine at Orono, Orono, ME 04473

[2] Department of Mathematics and Statistics, University of Maine at Orono, Orono, ME 04473

Introduction

Existing neuronal morphology data has been obtained using many different neuron tracing systems. Generally, these systems represent neuronal structure using a collection of points and connections in 3 dimensions. The different tracing systems use a variety of representations and vocabularies to describe the connectivity between points based on different conceptual models for the morphology data. In each case, the generated files are flat text files so that cell structure is not available explicitly without reconstructing the cell. Consequently, each tracing system requires a different tool for reconstructing neuronal data for use in spatial analysis or simulations. There has been growing interest in defining a common data format for describing and exchanging neuronal cell and model information using Extensible Markup Language (XML) technology ([1] and [4]). This work addresses the use of an XML application, MorphML, as a common data format for neuronal morphology data.

MorphML: An XML application for neuronal morphology data

XML provides a portable format for computer documents ([2], [5], [6], and [7]) where data is surrounded by text "markups" that describe the structure and semantics of the data. That is, the important details about the structure and the content of the document are

explicitly described by tags. XML also allows developers to define tags as they are needed. For the purpose of interoperability, researchers in the same domain often agree to use a particular vocabulary for the markup, resulting in a set of tags known as an XML application. The markup permitted in a particular XML application can be documented using a schema language such as Document Type Definition (DTD) or the W3C XML Schema Language.

The XML application that we have developed for the storage of neuronal morphology data provides a markup for representing digitized morphology files. These XML documents must have the same format regardless of the syntax of the original digitized morphology files. Consequently, our chosen vocabulary is an abstract representation language that is independent of any particular digitization syntax. Ultimately, these XML documents will be stored in databases, viewed by visualization tools, and passed to analysis and simulation software. Therefore, the data format has not been modeled on the internal structure of any specific application software. For all of these downstream applications, neuronal cell structure information is essential. Thus, in addition to storing original information about the digitization, these XML documents provide reusable information about cell structure as shown in Figure 1. The application has been well documented since XML documents will mainly be used to share neuronal cell morphology data among neuroscience research labs.

Both a schema and a DTD have been developed for neuronal morphology data and are available at http://www.umemat.maine.edu/faculty/crook/MorphML. The schema,

Morphology.xsd, which is provided in XML Schema Language, provides a comprehensive and controlled support for extending a markup using the `extension`, `include`, `redefine`, and `import` elements. Every element in the schema is a named complex type, which facilitates data type reuse and application extension. To facilitate easy handling of white space by downstream application software, every element contains only complex content, and no element contains character data. The schema has been defined to accommodate many different types of cells, cell features, and other histological features as demonstrated in Figures 2 and 3. The complex types and elements shown in Figures 1 through 3 are only a subset of the available data types in MorphML.

Another XML application, NeuroML, has been developed to support discussion, development, and exchange of neuronal models ([1]). MorphML can be used separately as a stand-alone application for representing morphology data. However, it is also suitable for inclusion in NeuroML using the simple inclusion method. An element of NeuroML can be defined as `Morphology` so that it will be a container element that contains the definition of our schema. Both NeuroML and MorphML are defined syntactically using object-oriented Java classes. MorphML can also be incorporated into NeuroML by writing descriptive scripting code. The attributes `elementFormDefault` and `attributeFormDefault` are set to `qualified` to force every element from MorphML to be explicitly qualified with the namespace and avoid potential tag-name collision.

The tool nemo2xml

The ultimate goal of our work is to coordinate the many components of a simulation environment that can be used to store, reconstruct, model, and study networks of neurons from the cricket cercal sensory system. Interface modules are required to make these components work together as a system. The cricket neurons were digitized using the Nevin system and saved as Binary Branch Tree (BBT) files. Because these digitized morphology files generally contain a large number of data points, the corresponding MorphML documents must be created automatically by software tools. Therefore, a Java tool, nemo2xml, was developed to automate the generation of a MorphML document from one or more BBT files. Nemo2xml reconstructs the cell morphology in memory from the BBT file then traces through the structures and generates the correct tags for encoding in MorphML.

Many tools are available that can be used to validate documents generated by nemo2xml against the defined DTD or schema, such as XMLSPY, Topologi Schematron Validator, and Xerces. When programs that process MorphML documents are developed in the future, the parsers' API should be designed to validate documents.

References

[1] Goddard, NH, M Hucka, F Howell, H Cornelis, K Shankar, and D Beeman. 2001. Towards NeuroML: Model Descrition Methods for Collaborative Modelling in Neuroscience. Available at: http://www.neuroml.org

[2] Harold, ER, and WS Means. 2002. XML in a Nutshell. O'Reilly & Associates, Inc. Sebastopol, CA.

[3] Hines, ML, and NT Carnevale. 1997. The NEURON Simulation Environment. *Neural Computation* 9:1179-1209.

[4] Hucka, M, K Shankar, and D Beeman. 2000. Preliminary Definitions of the Representation Schemas Used in the Modeler's Workspace. Available at : http://www.genesis-sim.org/hbp/mws-rep/mws-rep.html

[5] Maruyama, H, K Tamura, and N Uramoto. 1999. XML and Java^{TM}. Developing Web Applications. Addison Wesley Longman, Inc. Reading, MA.

[6] McLaughlin, B. 2001. Java and XML. O'Reilly & Associates, Inc. Sebastopol, CA.

[7] W3C. 2000. Extensible Markup Language (XML) 1.0 (Second Edition). Available at: http://www.w3.org/TR/REC-xml
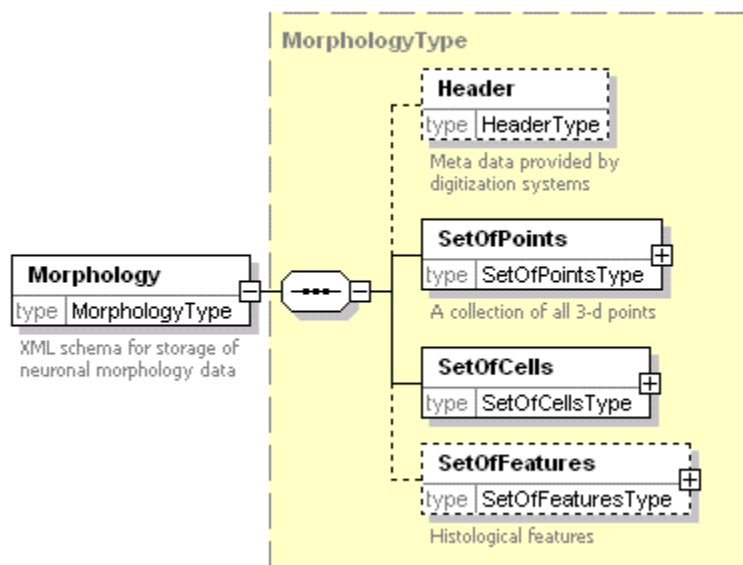
Figures

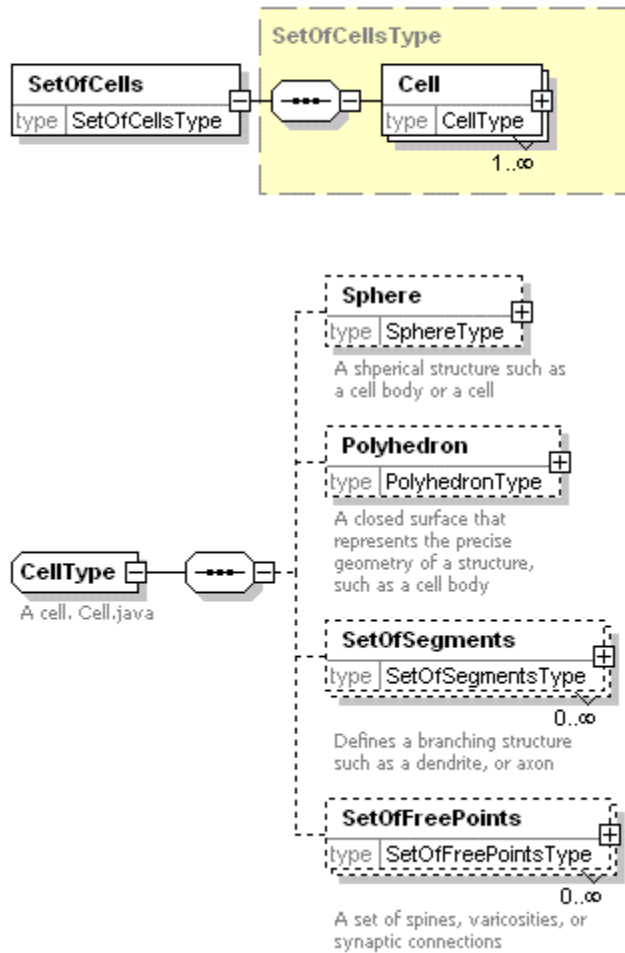Figure 1. Overview of the MorphML schema for storage of neuronal morphology data.

Figure 2. The complex type, **CellType,** has a number of subelements that can be used to describe cell morphology at different levels of complexity.
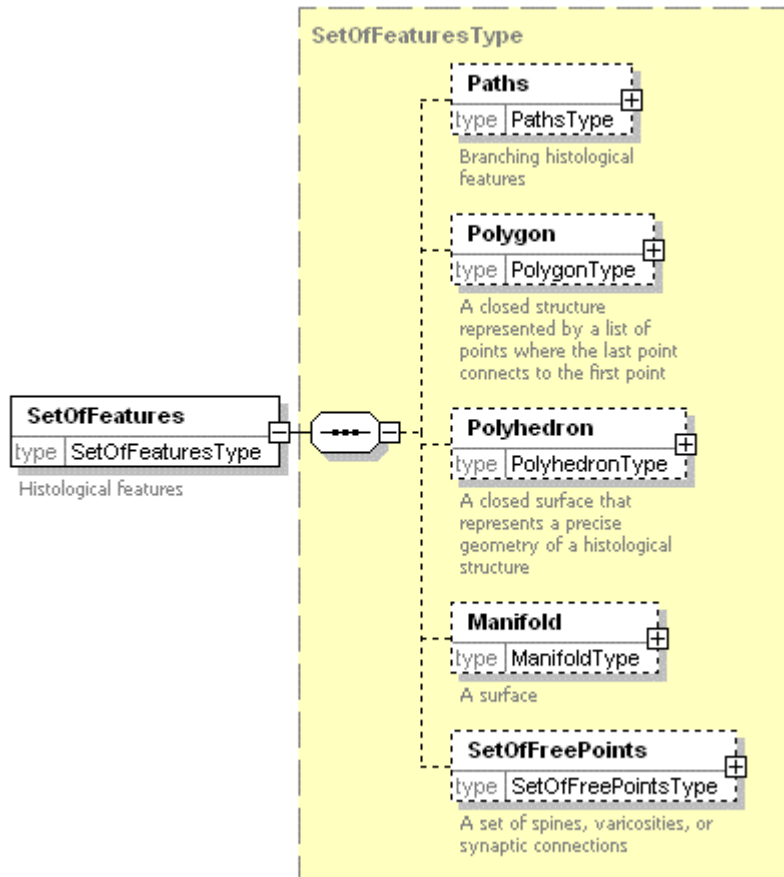
Figure 3. Additional elements have been developed for describing other anatomical and histological features such as spines, synaptic locations, and fiducial structures.