Figure 5: The neuron geometry (a) is based on an artistic CAD rendering generated with the public domain version of the software Blender (http://www.blender.org). In order to conduct simulations in this geometry, the model was exported in the STL surface mesh format, imported into the open-source meshing package Gmsh \[20\], where the boundary was re-parametrized and the domain subsequently meshed with a volume mesh in 3D. The resulting mesh was then converted into a Comsol Multiphysics 3.5a model to serve as a geometry description for the URDME model. Assembly of active transport jump rate constants are conducted by URDME on the unstructured mesh shown in (b). For a mathematical background on how to obtain these constants on the unstructured mesh, see \[26\]. URDME's capability to use an unstructured mesh made up of tetrahedral and triangular elements is of vital importance in order to be able to resolve the complex geometry of the neuron.

In order to setup this simulation in URDME, a Matlab function for the velocity field modeling the average orientation of the fibers at any point in the domain needs to be provided. Obviously, specification of this velocity field requires biological knowledge. The ability to work in the Matlab environment greatly simplifies parametrization of the velocity field. Since this geometry was given as a surface mesh, which is also often the case when the domain is obtained from cell imaging, we have no analytical expression for the parametrization of the geometry to rely on. In this example we want the velocity field to trace the axon and dendrite structures. To achieve this, we first compute surface normals to all triangles on the surface of the neuron. An interpolation table containing vectors with base in the centroids in the triangles of the surface mesh and pointing in the direction of suitably chosen reference points was thus constructed. For simplicity, we only used two different reference points, one near the center of the cell body and the other beyond the axon terminus along the long axis of the axon. The smoothness of the velocity field can easily be improved by adding more reference points. For any point inside the domain, we evaluate the velocity by nearest neighbor interpolation using the interpolation table. From this description of the microtubule network and the information about the mesh, utility routines available as add-ons to the basic URDME package can be used to assemble jump rate constants to be used in the definition of the stochastic transport process in much the same way as for diffusion \[26\]. This procedure may seem complicated at a first glance, but can be performed quite easily in Matlab using built-in utility routines. The model files required to run this example can be found in Additional File 5.