Topologic

INTRODUCTION

Topologic v. 0.7.x

January 2019

TABLE OF CONTENTS

Our Story

As the English proverb goes "necessity is the mother of invention". Topologic grew out of a need to build lightweight models that are adept at representing space rather than just building fabric. Dr. Wassim Jabi, Reader at the Welsh School of Architecture, Cardiff University, had been researching parametric design thinking and the role of building performance simulation in the early design stages. He attended an Autodesk workshop in London, presented by Dr. Robert Aish, during which the concept of non-manifold topology was introduced. Impressed by its potential, Dr. Jabi used this capability in his research and published a paper on the subject. Unfortunately, when Autodesk combined Designscript with Dynamo it decided to remove the non-manifold topology capability from the final product. Needing to continue his research, Dr. Jabi first turned to Autodesk's 3ds max as a substitute. However, its non-manifold capability was limited. So, he approached Dr. Aish, who had by then left Autodesk and joined the Bartlett School of Architecture, with an idea to collaborate on a research proposal to fully investigate the capability of non-manifold topology and build an open-source software library that would work with visual data flow programming systems such as Dynamo and Grasshopper. The proposal was funded by the prestigious Leverhulme Trust in 2016 for a three-year, £300,000 project based in Cardiff University in collaboration with University College London. Topologic is the result of this three-year effort.

Project Team

- Dr. Wassim Jabi, Cardiff University, Project Leader
- Dr. Robert Aish, UCL, Co-Investigator
- Dr. Simon Lannon, Cardiff University, Co-Investigator
- Dr. Aikaterini Chatzivasileiadi, Cardiff University, Research Associate
- Dr. Nicholas Mario Wardhana, Cardiff University, Research Associate

Introduction

Topologic is an open-source software modelling library enabling hierarchical and topological representations of architectural spaces, buildings and artefacts through non-manifold topology (NMT). Topologic is designed as a core library and additional plugins to visual data flow programming (VDFP) applications and parametric modelling platforms commonly used in architectural design practice. These applications provide workspaces with visual programming nodes and connections for architects to interact with Topologic and perform architectural design and analysis tasks.

Topologic is implemented using a multi-layer software architecture. At the lowest layer, we use Open CASCADE, an open-source NMT geometry software development kit (SDK) that provides data structures and modelling algorithms for 3D solid structures. We also use ShapeOp, an open-source SDK for surface planarization. Classes and methods in these two SDKs are encapsulated in the second software layer containing the TopologicCore and TopologicSupport libraries, written in C++. TopologicCore implements the core topologic classes and methods using an object-oriented programming (OOP) approach while TopologicSupport provides added utilities as needed. Above this layer, we implemented an interface layer, written in the .NET C++ / CLI language, that connects the core and support libraries to the host geometric editor or visual data flow programming application. At present, this layer (Topologic[VDFP]) has been written for Autodesk Dynamo software and is thus named TopologicDynamo. Work is underway to implement a version for McNeel Rhino/Grasshopper 3D (TopologicGH). Additionally, we envisage plug-in developers will use this layer to develop domain-specific applications. By strongly separating the code written for different platforms, this architecture ensures high modularity and code readability. In addition, the software can be easily extended to other platforms by writing a small library using the platform's conventions in the upper layer to encapsulate the core library.

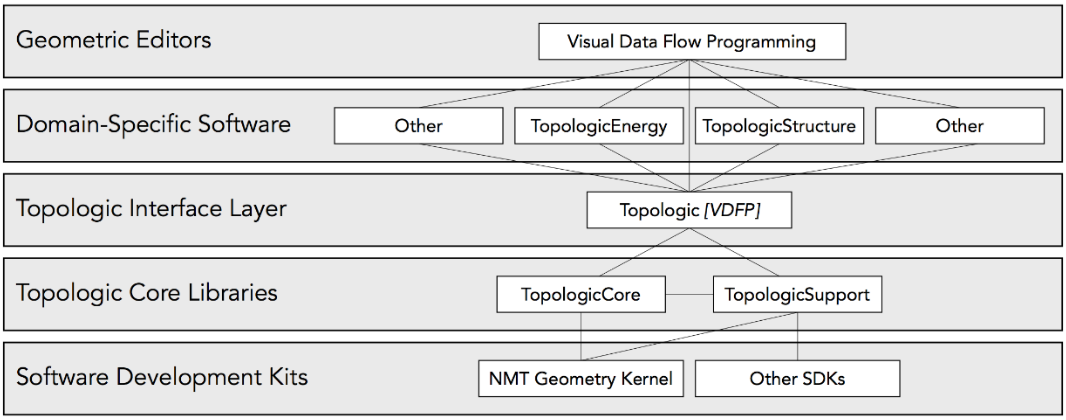

Figure 1. Topologic multi-layered software architecture.

Build, Connect, Analyse

In a traditional 3D modelling environment, solid objects (e.g. polyhedral) are said to have a 2-manifold boundary. If one imagines the boundary to be flattened and made infinite, then each point on this boundary is completely surrounded by other points on that 2-dimensional boundary. Examples of 2-manifolds include the surface of a torus, a sphere, or a prism. More importantly, each point on the boundary of a 2-manifold solid divides the modelling space into two regions, the solid material inside the boundary and the void of the outside world. Topologic aims to enhance the computational parametric design of architectural space using a novel technique based on the mathematics of non-manifold topologies (NMT). A non-manifold topology is defined as the condition at which a point on the boundary does not divide the modelling space into two regions. Practically, non-manifold geometric models can be defined as combinations of vertices, edges, surfaces and volumes. Contrary to traditional solid geometry boundary representation, NMT allows for and consistently represents any combination of these elements within a single entity. Conversely, traditional boundary representation struggles with representations where a surface divides the interior of a polyhedron, an edge is shared by more than two surfaces or ones that combine an isolated vertex, edge, surface and a solid in one representation.

Topologic is well-suited to create a lightweight representation of a building as an external envelope and the subdivision of the enclosed space into separate spaces and zones using zero-thickness internal surfaces. Because Topologic maintains topological consistency, a user can query these cellular spaces and surfaces regarding their topological data and thus conduct various analyses. For example, this lightweight and consistent representation was found to be well-matched with the input data requirements for energy analysis simulation software. Because Topologic allows entities with mixed dimensionalities and those that are optionally independent (e.g. a line, a surface, a volume) to co-exist, structural models can be represented in a coherent manner where lines can represent columns and beams, surfaces can represent walls and slabs, and volumes can represent solids. In addition, non-building entities, such as structural loads can be efficiently attached to the structure. This creates a lightweight model that is well-matched with the input data requirements for structural analysis simulation software. With Topologic, you can also build dual graphs and conduct space syntax analysis efficiently using lightweight models with represented rooms as spaces with zero-thickness dividing surfaces with imbedded apertures (e.g. doors).

Class Hierarchy

TopologicCore contains the following main classes:

- **Topology**: A Topology is an abstract superclass that stores constructors, properties and methods used by other subclasses that extend it.
- **Vertex**: A Vertex is a zero-dimensional entity equivalent to a geometry point.
- **Edge**: An Edge is a one-dimensional entity defined by two vertices. It is important to note that while a topologic edge is made of two vertices, its geometry can be a curve with multiple control vertices.
- **Wire**: A Wire is a contiguous collection of Edges where adjacent Edges are connected by shared Vertices. It may be open or closed and may be manifold or non-manifold.
- **Face**: A Face is a two-dimensional region defined by a collection of closed Wires. The geometry of a face can be flat or undulating.
- **Shell**: A Shell is a contiguous collection of Faces, where adjacent Faces are connected by shared Edges. It may be open or closed and may be manifold or non-manifold.
- **Cell**: A Cell is a three-dimensional region defined by a collection of closed Shells. It may be manifold or non- manifold.
- **CellComplex**: A CellComplex is a contiguous collection of Cells where adjacent Cells are connected by shared Faces. It is non- manifold.
- **Cluster**: A Cluster is a collection of any topologic entities. It may be contiguous or not and may be manifold or non- manifold. Clusters can be nested within other Clusters.

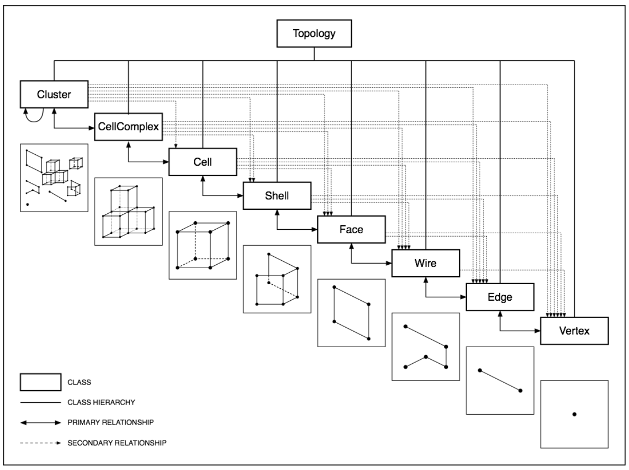

Figure 2. Topologic class hierarchy.

The Leverhulme Trust

This project is funded by a Leverhulme Trust Research Project Grant
(Grant No. RPG-2016-016)

CONTACT US

Web: topologic.app

Email: info@topologic.app

Phone: +44 (0) 29 2087 5981

Welsh School of Architecture
Cardiff University
Cardiff CF10 3NB
United Kingdom