2013 IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops

Tile Assembly System

A software package for Tile-Based Algorithmic Self-Assembly*

Tyler Fochtman
Department of Computer Science
University of Arkansas
Fayetteville, Arkansas
tfochtma@uark.edu

Matthew Patitz
Department of Computer Science
University of Arkansas
Fayetteville, Arkansas
patitz@uark.edu

I. Introduction

Tile Assembly System (TAS) is a free, open-source, and cross-platform application that enables research and education in the field of tile-based algorithmic self-assembly through a full development and simulation environment for the abstract Tile Assembly Model (aTAM), the kinetic Tile Assembly Model (kTAM), and the 2-Handed Assembly Model (2HAM) [1]. There is also a developed wiki site page, selfassembly.net, which has information on algorithmic self-assembly as well as tutorials for TAS [2].

II. Overview of Demonstration

Using TAS, this demonstration will provide a basic introduction to the three most widely used models of tile-based self-assembly: the abstract Tile Assembly Model (aTAM), the kinetic Tile Assembly Model (kTAM), and the two-handed Tile Assembly Model (2HAM). Each model provides a defined process by which sets of relatively simple objects (e.g. tiles) can autonomously combine into structures of greater complexity, i.e. self-assemble. The general purpose of the aTAM is to provide a framework which can be used to discover the mathematical properties of tile-assembly. The kTAM builds upon the aTAM and provides an accurate model by which the nuances of chemical kinetics and probabilistic assembly (including errors) come to bear on what is feasible to self-assemble, and as such has helped guide laboratory experiments as well as predict results. The 2HAM is like the aTAM in that it ignores kinetics, however, each iteration of a 2HAM simulation produces new sets of self-fabricated supertiles that potentially become substructures of the next generation of supertiles, thus allowing arbitrarily large assemblies to combine in pairs.

Within the demonstration, the unique properties of each model will be exhibited by tile-based counters, i.e. algorithmic assembly systems which inherently execute binary counting. The assembly of predefined tile-based structures and patterns are as reliant on measurement as is the construction of a brick and mortar building. The challenge of constructing objects at the nanometer scale is that we can't explicitly move the materials where we want them. Instead, we must algorithmically guide the placement of materials through the self-assembly of modules such as counters. Counters can then act as scaffolding from which other portions of the structure can be assembled. This demonstration will provide examples of counters in the aTAM, kTAM and 2HAM.

III. About TAS

In 2009 Dr. Matthew Patitz released the first version of TAS. Since 2009 TAS has maintained its relevance as a research tool through consistent upgrades and refactoring of the underlying codebase. TAS is a free application released under the GNU GPL license. The source code for TAS is available for download and is written in C++. TAS is also a cross-platform application that operates on Windows, Linux and OS X.

The user-friendly GUI allows veterans in the area of research to quickly develop tile sets, but is also intuitive enough for newcomers to grasp. There are three main windows available to the user: the simulation space, the tile-set editor, and the overview window. The simulation space changes depending on which mode (aTAM, kTAM, or 2HAM) the application is in, presenting options specific to each model. The tile-set editor window facilitates designing, building, and editing tiles. Within this window the user can define new tiles based on the respective edge glue strength and edge label. Users can also color specific tiles to draw out patterns of the tile-assembly. The overview window is a small window that allows for quick navigation of large tile assemblies.

Each simulation mode has unique configuration options built into the GUI that allow TAS users to dynamically manipulate and glean insight from the simulation space. For example, simulations are cached and can be rewound for inspection, and while in the 2HAM simulation mode the number of supertiles for each tile count are kept in a list and this list is updated with each iteration. Also in the 2HAM, the user is able to filter the viewable set of supertiles based on size parameters. For example if the user is only interested in studying a range of tiles, they may exclude all but this range from view within the simulation space. Another important 2HAM feature is the ability to permanently remove subsets of tiles from the simulation space.

For researches and experienced users of TAS there are several notable features of the application. TAS has the ability to automate the simulation of tile assemblies through the Batch processing mode. In Batch mode the user can run an arbitrary number of simulations and save desired

* This research was supported in part by National Science Foundation Grant CCF-1117672.

978-1-4799-5086-7/13 $31.00 © 2013 IEEE
DOI 10.1109/SASOW.2013.29

characteristics of those runs to a file for analysis. TAS can work with any tile-set size that the underlying computer hardware can handle, and has been optimized to make even large simulations over very large tile sets feasible and fast. Additionally, while TAS has its own simple and well documented file format for defining tile assembly systems (to facilitate external, programmatic generation of system definitions), it is also able to utilize systems defined for the other well-known simulator, xgrow.

REFERENCES

[1] M. Patitz, “An Introduction to Tile-Based Self-Assembly and a Survery of Recent Results”, Natural Computing, to appear.

[2] Tutorials and Video Demonstration of TAS: selfassembly.net/wiki/index.php?title=ISU_TAS_Tutorials