information to evaluators of collaborative systems. Using the results of the card sorting exercise I assisted in developing the trial taxonomy, by grouping the applications together, and by naming each group to see if each set would be a useful category for collaborative systems.

FUTURE

The next step will be to carry out a formal experiment to validate the refined heuristics. Participants will be given collaborative problems and will be asked to state which heuristic each violates. Agreement of the subject's opinions can help to indicate whether or not the heuristics are understood and as a result can be applied to collaborative systems.

ACKNOWLEDGMENTS

Thank you to the director of the URSP program at Saint Anselm College, Dr. Kathleen Flannery, for giving me the opportunity to conduct my research. I would like to acknowledge the Computer Science Department at Saint Anselm College, chaired by Dr. Paul Kenison, and my fellow students Elizabeth Maestranzi and Jonathan Madore. Thank you to Dr. Marian G. Williams, my external advisor in the URSP program, and the Human-Computer Interaction Research Group as UML. I would like to also thank my advisor, Dr. Carol Traynor, for her unconditional encouragement and support and special thanks to Ms. Jill Drury for her dedication and friendship.

VISUALIZATION OF AN ALGORITHM FOR CONVEXIFYING A SIMPLE PLANAR POLYGON WITH RIGID MOTIONS

Elif Tosun
Computer Science Department
Smith College
Northampton, MA 01063-0100
(413) 585-7149
etosun@cs.smith.edu
Faculty Advisor: Ileana Streinu, streinu@cs.smith.edu

The Carpenters Ruler Problem asks whether a simple planar polygon can be moved continuously to a convex position, so that the edge lengths and simplicity are preserved along the way. The main goal of our research is to visualize the recent algorithmic solution of [1] to this problem. The algorithm is based on the idea of adding extra bars to the polygon to form a

JCSC 16, 4 (May 2001)

mechanism with one degree of freedom, derived from a pseudo-triangulation. The mechanism is then moved continuously until two edges align, at which point a local change produces another mechanism and process continues for a finite number of such steps.

We visualize the algorithm using several tools: the interactive geometric software Cinderella and OpenGL. We start with the initial drawing of a simple planar polygon with fixed edge lengths in Cinderella. As a preliminary version, for consecutive steps of the algorithm until the polygon is convexified, we create frames with the current position of the vertices, so that when displayed successively they would create an animation (Fig.1). Then to get the final graphical animation movie, we feed the Cinderella files for each frame into a C++ program that uses OpenGL for rendering.

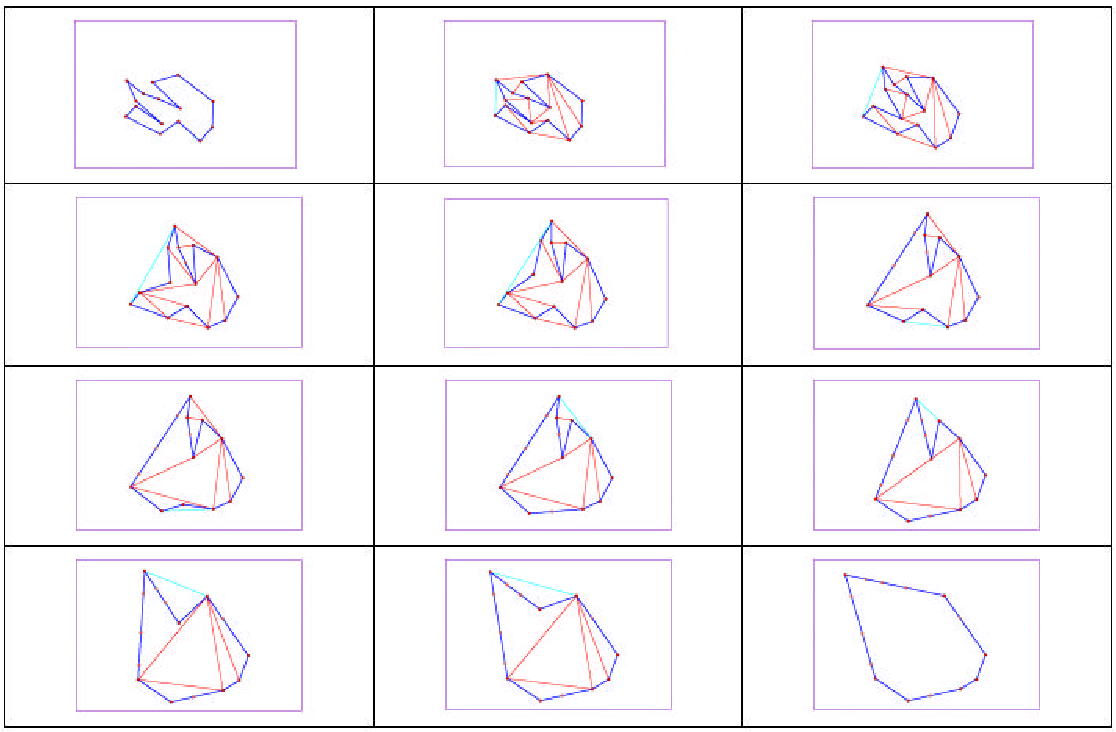

Fig1: Some frames created using Cinderella for the preliminary animation.

The challenge in visualizing this algorithm comes from the desired continuous motion. When an edge is rotated around a vertex, the other vertices follow various algebraic curves. To implement this algorithm, we would have to rely on algebraic approximations and that can cause error accumulation. This would be a major problem in this algorithm since it's crucial that the lengths of the edges of the polygon stay fixed throughout the motion. The software Cinderella makes good approximations keeping the edge lengths fixed. However, there are restrictions in the uses of Cinderella: It is not possible to visualize every rigid simple planar polygon with fixed edge lengths using this software, but the underlying reasons are deep mathematics.

The next step of this project is the implementation of the algorithm, starting with the combinatorial parts, which don't rely on algebraic computations. The preliminary version of the visualization of the algorithm, which is an animation, can be found at the following

URL: http://cs.smith.edu/~streinu/Research/Motion/motion.html

REFERENCE:

[1] Streinu, Ileana. "A Combinatorial Approach to Planar Non-Colliding Robot Arm Motion Planning" Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS), Redondo Beach, California, 12-14 November 2000, pp. 443-453. IEEE Computer Society Press, isbn 0-7695-0850-2.

LAYERED MEMORY USING BACKWARD-CHAINING

Joshua Gay
Department of Mathematics and Computer Science
Suffolk University; Boston, MA 02108-2770
JoshuaGay323@aol.com
Advisor: Prof. Louis W. G. Barton
louis.barton.je.77@aya.yale.edu

This is a progress report on my research project to design a model for *layered memory* in an intelligent agent. I am using the cognitive model of human memory [4] as the design reference. It defines three layers: sensory information storage (SIS); short-term memory (STM); and long-term memory (LTM). SIS processes sensory inputs and motor skills. It is fast but 'brittle' in being difficult to reprogram. It can be implemented as neural networks. STM is the seat of rationality and goal-setting. It is relatively fast, but its storage is small (perhaps seven semantic 'chunks'). It can be implemented as a knowledge-base (KB). LTM stores vivid sensory impressions, heuristic rules, and syntactic knowledge. Its storage is large but slow; reconstruction of STM state from LTM may be unreliable. It can be implemented as a database. The focus of my research at this point is the STM layer. A key feature of my STM model is backward-chaining inference. I present here a prototype of my *layered memory* model and the important points I have learned in implementing the STM layer.