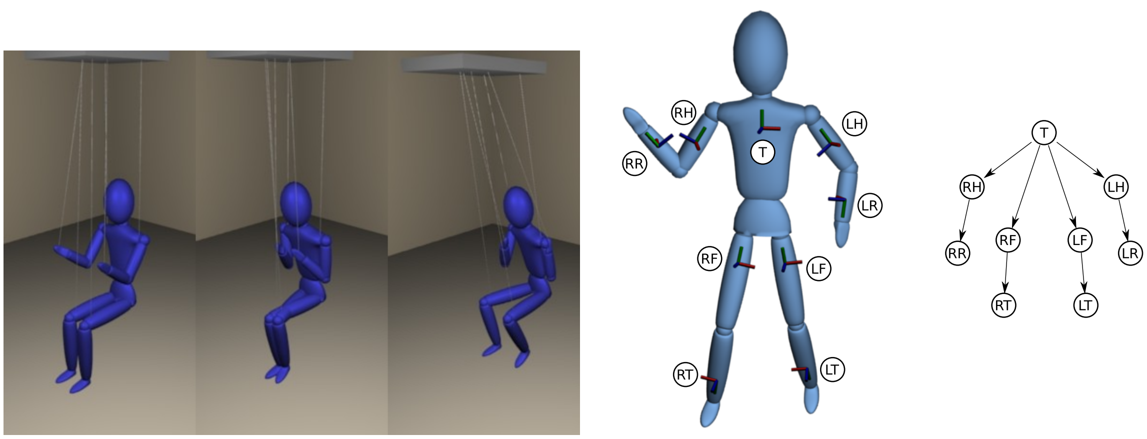
The Automated Marionette Project

Elliot Johnson
Todd Murphey
University of Colorado at Boulder

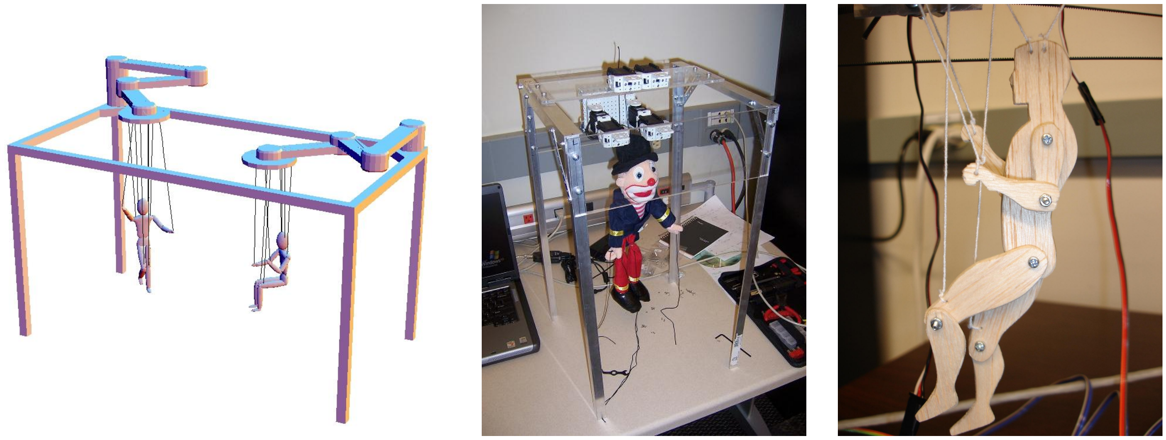
Puppet choreography is a highly-developed language for controlling mechanically complex marionettes. It has evolved over centuries into a largely standardized form that allows puppeteers to address issues that arise as a result of the complex systems with which they are working. As such, the standardization of the choreography can be thought of as the puppeteers' response to complexity. The research, in collaboration with Magnus Egerstedt and Patrick Martin at the Georgia Institute of Technology, focuses on understanding how puppeteers address complex tasks in their choreographic descriptions of plays and using that understanding to solve questions of importance to computer science and engineering. These goals will be achieved by creating an automated puppet play, which will use insights about puppet choreography to implement embedded control of mechanically complex marionettes engaged in complex coordination tasks.



(a) (b)

Figure 1: (a) Simulation of a string marionette and (b) a marionette mechanical system and its associated graph representation. For a specific example of a string marionette, see *http://puppeteer.colorado.edu*.

In order to automate a puppet play, there are three key technical hurdles that must be addressed: real-time embedded motion control; strategically handling the complexity associated with coordination of high degree-of-freedom systems; and computer animation and simulation. These three areas are of critical interest to the computer science community, entirely aside from their application to puppetry. In most current engineering applications real-time embedded control of physical systems only addresses comparatively simple systems, such as single degree-of-freedom mechanisms. Puppets often have more than forty degrees-of-freedom (as seen in Fig. 1, where the algebraic graph on the right is used to represent the puppet on the left), indicating that the associated embedded control must take their complexity into account. To facilitate understanding how puppet choreography addresses this problem, the puppetry artists will teach the researchers how to manipulate puppets and use choreography. Other additional complexity issues must be resolved in order to successfully put on a play with multiple puppets. These include communication between the embedded controllers, dynamic resource allocation, and synchronizing motion. The proposed



(a) (b) (c)

Figure 2: (a) A *Mathematica* graphical representation of the mechanical system we plan to eventually build; (b) a first-generation 3D experimental puppet platform at the Georgia Institute of Technology that cannot spatially translate in the environment; (c) a first-generation 2D experimental puppet platform at the University of Colorado that can spatially translate in its environment.

automated puppets will use a programming language that *directly mimics* the constructions used in puppet choreography. The puppeteers will additionally provide feedback on the implementation, indicating whether the implementation is not simple enough, is too simple, is abstracted improperly, or needs to be corrected in other ways. The goal will be to formalize how puppeteers strategically deal with complexity, task description, deadlock, problem specification, etcetera, in the context of puppetry. Lastly, the ability to simulate and visualize complex mechanical puppets is essential for testing and validating a potential choreographic specification of a play. However, current fast animation techniques use heuristics to resolve motions involving mechanical contact and are not amenable to control design and synthesis, so new animation techniques will be developed that are amenable to these tools, and will lead to tools that the puppeteers themselves will be able to use in the design of plays. *Hence, the project's goal is to use puppetry and puppet choreography as a source of insight into pressing computer science and engineering problems.* Experimental implementations such as those seen in Fig. 2 will be used to validate the techniques as well as put on a play at the Atlanta Center for Puppetry Arts.

Acknowledgments

We gratefully acknowledge the National Science Foundation and the Creative IT program for their support of this project under award IIS-0757378.