ArtiSynth White Paper

ArtiSynth project-related information, including downloads, documentation and contact information can be found at the official ArtiSynth website at: www.artisynth.org

---

Abstract

Our group is developing a software tool that allows modeling the human vocal tract in three dimensions and simulating its dynamics. We further provide acoustic components that can be used to synthesize speech based on the given vocal tract model and its shape. We call our tool ArtiSynth – short for ARTIculatory speech SYNTHesizer. This white paper provides an overview of the project. The intended audience for this paper is researchers in speech-, health- and engineering-sciences.

Introduction

The short-term successes of traditional formant-based and acoustic-based speech synthesis techniques have largely overshadowed articulatory synthesis. While these techniques have met with success in some domains, they have reached the point of diminishing returns, and have still not proven able to produce natural-sounding speech from text, nor to synchronize speech with animated faces for advanced speech interfaces. Articulatory speech synthesis, in contrast, promises to provide the necessary platform to overcome these problems and become the next major advance in speech synthesis technology. To date, however, research in 3-dimensional articulatory synthesis has focused mostly on modeling sub-components of the overall vocal tract. What is needed is a complete platform to allow individual vocal tract subcomponents to be integrated and tested within the context of a working articulatory speech synthesizer that utilizes the best available technologies for the rest of vocal tract. Creating this platform is the goal of this project.

The main product of this project is a fully functioning 3-D model of the vocal tract capable of synthesizing speech built upon a flexible, modular and extensible infrastructure. This infrastructure is based on an open-source standardization framework to allow easy extension and adaptation by the research community, providing a tangible repository of research knowledge. Out of the box, it will provide a default articulatory speech synthesizer incorporating the best available technology. We have focussed our efforts on the infrastructure needed to provide a modular, extensible synthesizer using pre-existing, proven techniques. This scope has eliminated the need to develop new models of the vocal tract or new synthesis techniques in the short-term. Instead, initially, our main contribution is the "glue" that allows research in each component to be added in as it progresses. As the infrastructure develops, we are adding new pieces of research with the final aim of providing a state-of-the-art vocal tract model capable of producing speech. Upon this infrastructure, we envision exploring new directions in aero-acoustics, data extraction, and vocal tract tissue modeling by the research community.

Our Mission Statement: Design and implement ArtiSynth, a modular software system for synthesizing speech based on vocal tract anatomy.

Our team is developing ArtiSynth and model components that have not been developed or integrated sufficiently by other groups. This includes (a) an aero-acoustics module, (b) a framework for connecting different types of models to each other and (c) methods for incorporation of real speech production data. Our efforts are focusing on the following main areas:

- Open source framework
- Simulation of 3d vocal tract dynamics with different modeling methods
- Aero-acoustic modules and other speech synthesizers
- Vocal tract data collection
- Vocal tract visualization
- Validation and analysis with experimental framework,
- Data extracted from image and tracking data

Who is Artisynth for?

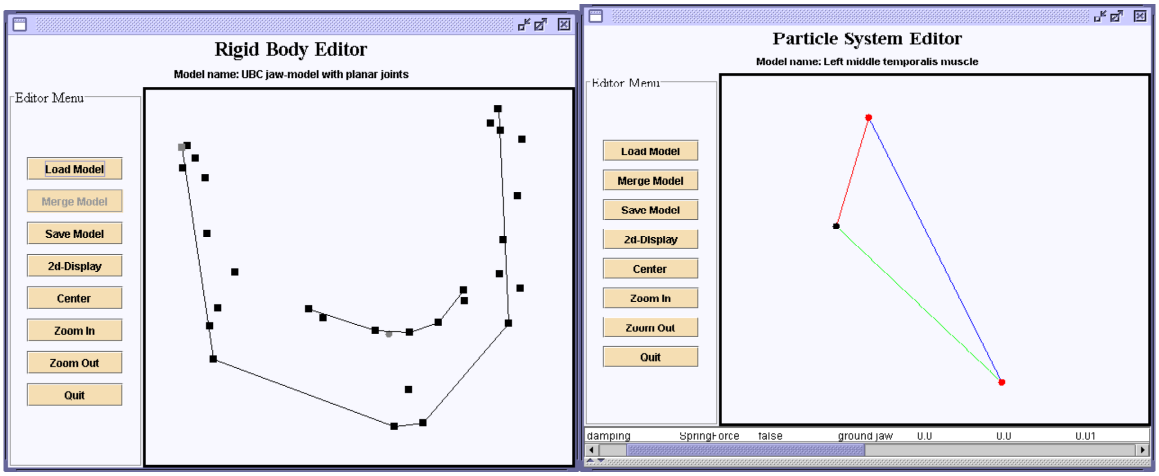
Artisynth is intended for speech and vocal tract anatomy researchers and practioners from fields such as linguistics, acoustic engineering, computer science, dentistry, cognitive science and oral surgery. We coarsely divide our user base into end-users and developers.

For end-users, we provide an easy-to-use graphical user interface for manipulating existing vocal tract models and speech synthesis models. For example, for a linguist with ultrasound data of tongue movements for some sound, he can compare tongue tractories with that of a default vocal tract model supplied with Artisynth. The interface allows the end-user to modify model parameters to match a particular speaker as well as adjust various speech parameters to hear the sound produced.

For developers, we support two levels of control: interpreter and Java level. We use the jython interpreter for quick prototyping, model development, control the simulation and modification of simulation settings. It can also be used as a 'glue' language to interface with other applications. Internally, Artisynth is structured as a toolkit with modular components that can be reused and modified in Java by developers. The open-source nature provides complete access to the class libraries that make up the core of Artisynth.

Highlights of Artisynth

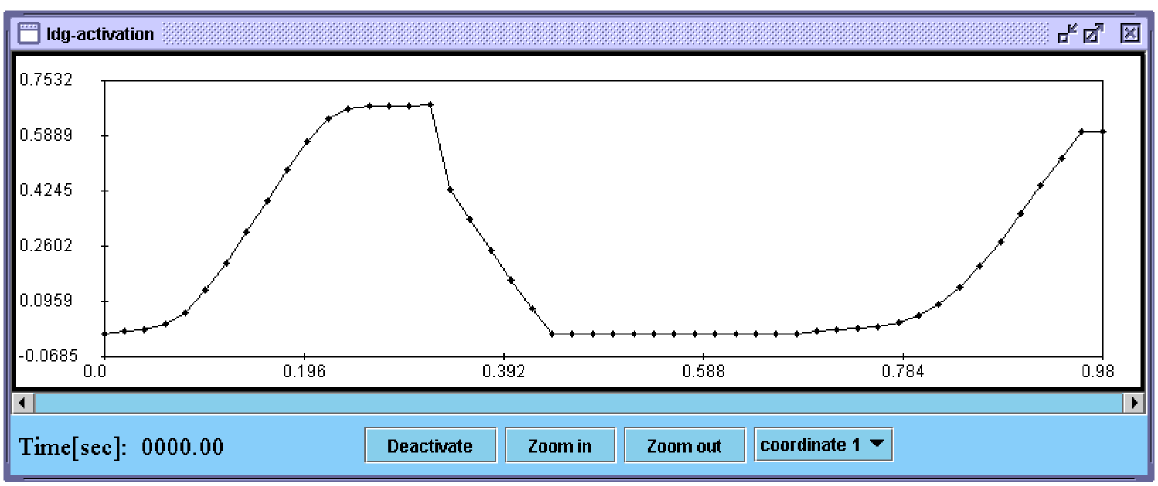
ArtiSynth provides a large variety of functions that allow researchers to create their own model of a complete vocal tract or just a part of the vocal tract, e.g. tongue. Alternatively, users can load a pre-stored work-space that includes models, connections, probes and simulation settings. The pictures below show examples of models that are available with ArtiSynth. The models are shown in marker-based mode, i.e. no meshes are drawn.



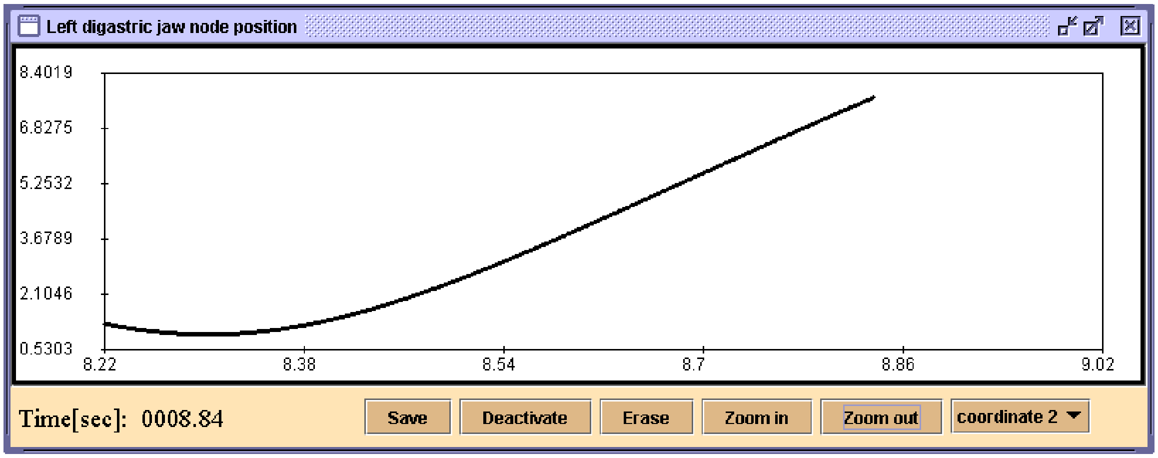
Left: Example of a jaw model that is represented as a rigid body. Squares define marker points that other models can connect to. Right: A model of a jaw muscle that connects the jaw with the skull. This model is based on the particle system component that comes with ArtiSynth. 3 particles and 4 muscles-specific non-linear forces are used in the model.

Users can connect their own models to already existing models through the use of marker objects. They can easily save the resulting model and make it available to colleagues. Alternatively, users can create their own vocal tract model by choosing and connecting models from a pre-existing model-collection of jaws, tongues, lips and other models.

And finally users can simulate the dynamics of the 3d human vocal tract models that they created or loaded. ArtiSynth provides objects called *probes* that have the ability to change model parameters during the simulation (input probes) and to monitor the state of specified variables (output probes). The pictures below show examples of input and output probes.



The plot above shows an example of an input probe. It represents a one-dimensional time series of activation for a jaw muscle. The maximum activation is 1.0 (100%). The probe is defined from time 0.0 to 0.98. Beyond time 0.98 the activation is considered to be 0.0.



The plot above shows an example of an output probe display. What is shown is the y-coordinate of the left digastric insertion point in the jaw model as it changes during the course of a simulation as a function of time (y-axis units are in millimeters).

What ArtiSynth does

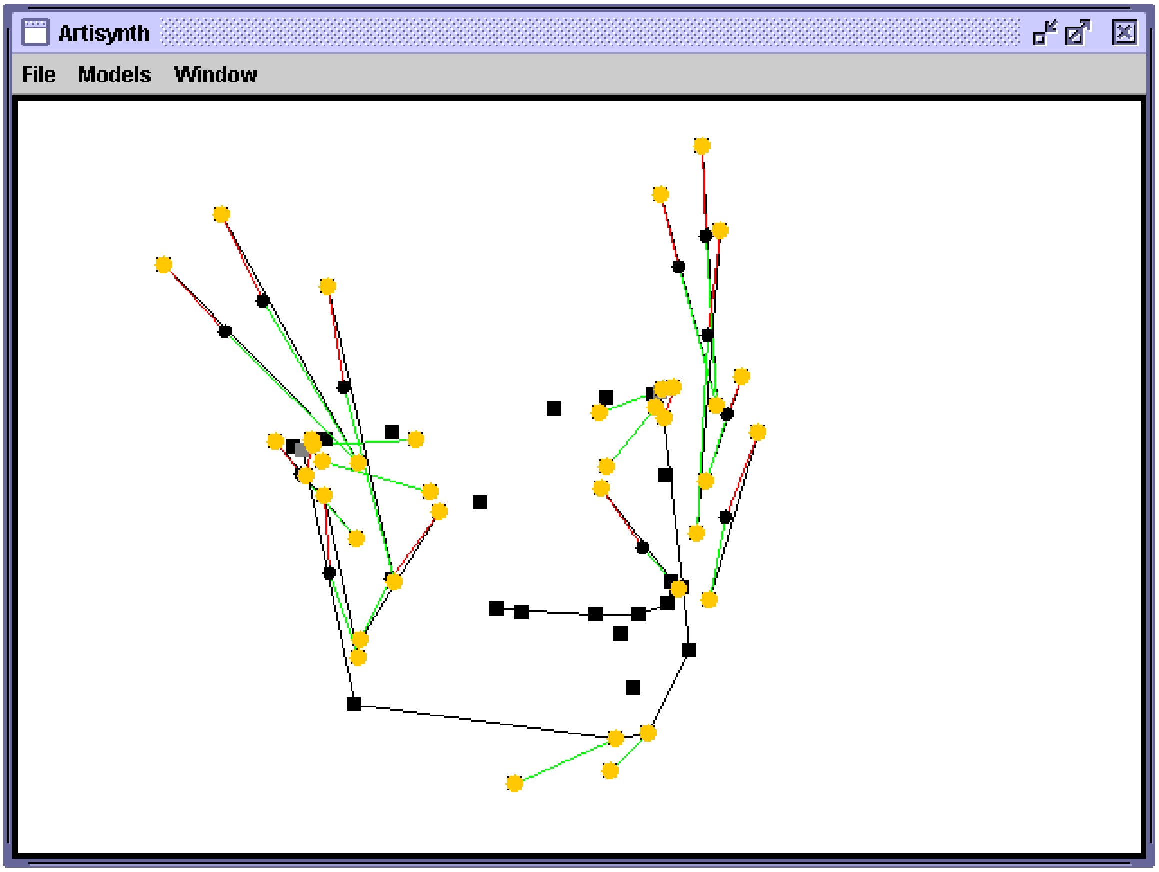
ArtiSynth is intended to be a modular, flexible infrastructure for researchers to test and compare existing and new models of articulatory speech synthesis using both physical simulation and data driven approaches.. It allows users to do the following:

- create and maintain a multi-structure dynamical anatomical vocal tract model
- simulate vocal tract dynamics based on the anatomical model of the vocal tract
- extent the existing component architecture (e.g. adding new model types)
- initially configure models with data
- use time-varying data to drive model dynamics (input probes concept)
- visualize models at any point in time (model snapshot and simulation output)
- export model parameters and state variables (output probe concept)

In addition to the above, the current version of ArtiSynth comes with a variety of models of anatomical vocal tract structures. They can be accessed through the Models menu on the main GUI page. An example of a model is the UBC-Jaw Model. It is shown in the screenshot below. Except for the Simple Model demo that includes simple examples of Particle Systems and Rigid Bodies all other models are related to the human vocal tract. Models included with ArtiSynth are:

- UBC-Jaw Model
- PCA-Tongue Model
- UBC-Jaw Model with Tongue
- UBC-Face Model
- UBC-Lip Model

Models in ArtiSynth are set up hierarchically. Thus, all models have a root node with sub-models being children and sub-submodels being children of children and so on. Users can also select the Empty Model which will delete the model that is currently in the WorkSpace and re-initialize the root node in the model tree so that new models can be added from scratch. This is also the configuration when ArtiSynth starts up.



The screenshot shows an example of the UBC-Jaw Model that consists of two rigid body models (jaw and skull) that are connected by 18 muscles. We are therefore looking at a model that is based on 20 individually connected models. The extended rigid body jaw model includes two planar constraints that represent simplified jaw joints.

What ArtiSynth will do

The ArtiSynth functionality will be extended with the following features:

- Aero-acoustic simulation
- Speech synthesis based on a variety of acoustics models.
- Combined physical animation and key frame animation
- Interoperatebility with Extraction and Registration of Animation, and analysis tools like modeling tools like Matlab
- Provide a research platform to develop and compare different modeling speech approaches
- Constrains can express physical laws such as gravity and collisions
- Landmarks will allow to match measured and simulated points in the anatomical structures
- Data-driven simulation

How to use ArtiSynth

ArtiSynth can either by run from a graphical user interface or from the jython command line. The command-line interface provides more functionality since every public method can be accessed directly but at the same time requires knowledge of jython and Java. The GUI provides the core functionality with the comfort of a mouse-click.

Users will mostly be interested in modeling and simulating vocal tract dynamics as well as using data to drive the dynamics or update the state of selected models. We provide a particle system and a rigid body model component that can be used to build models from scratch. The list below shows potential user scenarios:

- Create a new model from scratch, save it and send it to your colleagues
- Load a pre-existing model and run simulations to better understand the model
- Start with the UBC-Jaw Model and connect it to your tongue or lip model
- Create a complete 2d or 3d vocal tract model that contains skull, jaw, palate, pharynx, tongue and lips
- Load an existing model through the Models menu and demo it to students
- Experiment with an existing vocal tract model by adding specific input probes

Modeling

A model is build up by connecting individual models to each other. Models can have any shape or form as long as they inherit from the Model-class in the ModelBase-package. The user can build a model with editors of existing model components like the particle system or rigid body ones. They can also extend an existing model component and add specific functionality that they are interested in. This requires writing Java code though.

Simulating

The dynamics of a possibly connected model can be simulated. The simulation is handled by the Simulator package. The given model is simulated with a specified integrator like Runge-Kutta. Each integrator has its own set of parameters that can be specified.

Particle System

A ParticleSystem package is provided. We have used this model component extensively to model spring-mass system, a face model and jaw muscles. An editor is provided that allows configuring the Particle System and changing its parameters.

Rigid Body

A RigidBody package is provided. We have used this model component to model a human jaw and skull and we plan to use it for all rigid body parts in the human vocal tract. An editor is provided that allows configuring the Rigid Body.

Adding Components

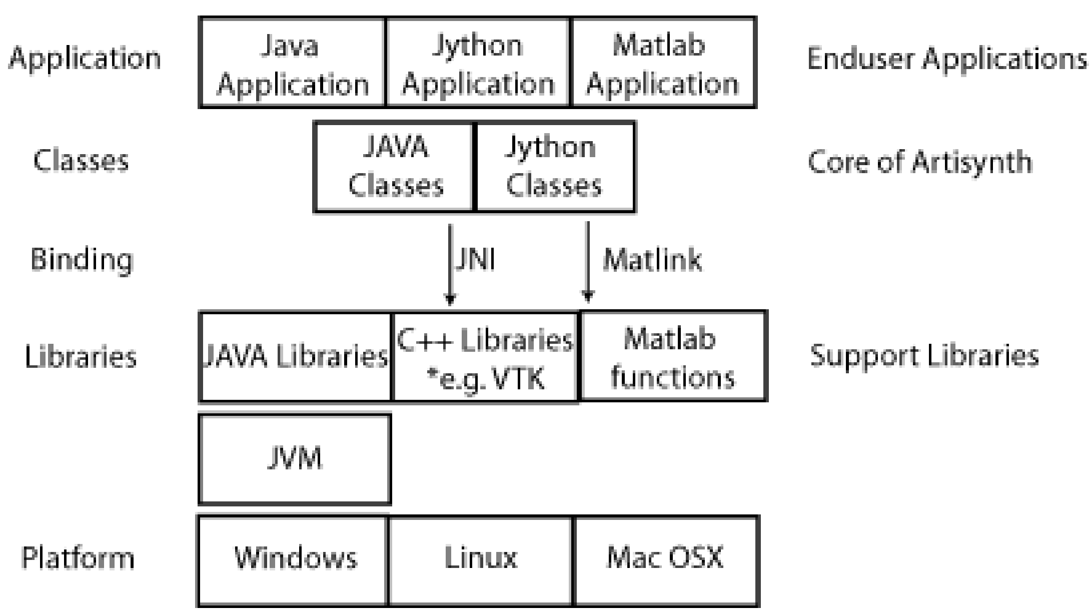
ArtiSynth's extendable architecture allows importing components. Users can write their own components, e.g. a new model type, or add several lines of code to include their existing code into ArtiSynth.

How ArtiSynth works

The core components of ArtiSynth are written in Java. Specific java-bindings allow code that is written in other languages to interface with ArtiSynth. We are currently developing a VTK-based renderer written in C++ that will extend the java-based renderer that is provided in the first version of ArtiSynth. Furthermore, we plan to connect ArtiSynth to Matlab-based models using Matlink as well as C/C++ code using JNI for numerical procedures in future releases.

ArtiSynth may be used as a stand-alone application as well as a toolkit used by other applications. The java core enables cross platform functionality while C++ might evolve more porting efforts.

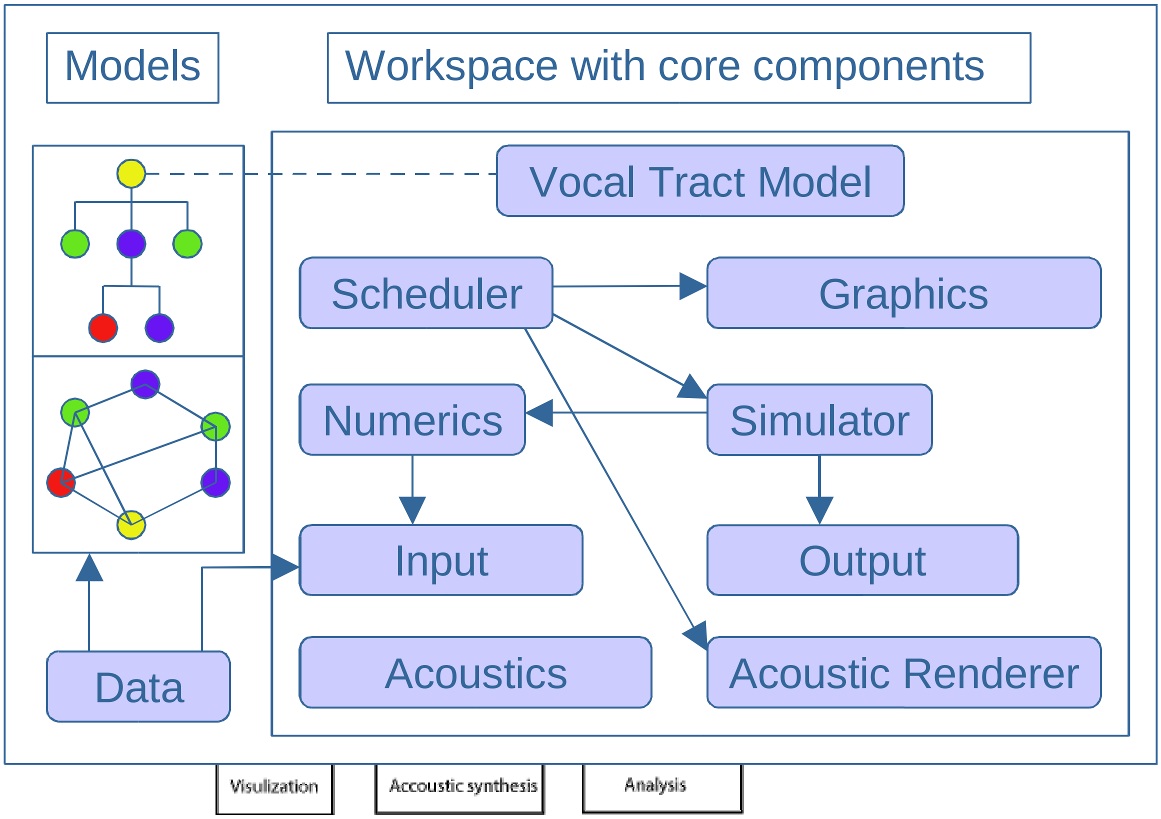
Artisynth Platform Diagram on Java 2 Version 1.4



Core components (implemented as java packages)

- WorkSpace: container for Models, Simulator, Scheduler and Graphics Renderer
- ModelBase: handles Models, Connections, Markers, Meshes and Probes
- Scheduler: responsible for scheduling integration calls and renderer updates
- Simulator: simulates model dynamics by: state integration or internal update
- Numerics: provides integration framework and several integration methods
- Render: contains the Renderer-base class and java-implementation of a renderer

Architecture diagram in class form



ArtiSynth Development process

Our software developers follow an iterative design and development process that tries to incorporate feedback from target users of ArtiSynth at an early stage. We work closely with researchers in linguistics and dentistry. For instance, a main part of our current vocal tract model is based on an existing jaw-skull-muscles model from dentistry that was successfully implemented with the ArtiSynth system. In the process of implementing the model we learned important lessons in model extensions, input probe functionality and model consistency.

Proposed workplan

Comment: Would be nice to have a tool that can create a roadmap-diagram. I looked at the grant proposal and we have a lot of items in the roadmap/timeline, some of which might or might not be important at the current time. I suggest that the project PIs get together and re-evaluate the roadmap and add it HERE. Ideally this is a nice diagram but a bulletin list will work, too.

Progress to date

We released ArtiSynth version ?? on ???, 2004. You can find full documentation and demo applets of this version on the ArtiSynth project page at: www.artisynth.org
This version performs the following main functions. It allows the user to:

- create and maintain a multi-structure dynamical anatomical vocal tract model
- simulate vocal tract dynamics based on the anatomical model of the vocal tract
- extent the existing component architecture (e.g. adding a new model type)
- initially configure models with data
- use time-varying data to drive model dynamics (input probes concept)
- visualize models at any point in time (model snapshot and simulation output)
- export model parameters and state variables (output probe concept)
- add: data collection, preparation, VTK, face-model, jaw-muscle-skull model, etc.

Team

We have an interdisciplinary research team. Current members are Sidney Fels (PI), Bryan Gick, Carol Jaeger, Alan Hannam, Eric Vatikiotis-Bateson, Florian Vogt, Oliver Guenther, Donald Derrick, Leah Vilhan, Kees van den Doel and John Lloyd.

Former team members and contributors include Kalev Tait , Ian Wilson, Rahul Chander, Charles Wilson and Maciej Gara.

Project-related information, including downloads, documentation and contact information can be found at the official ArtiSynth website at: www.artisynth.org

References