

# Non-Verbal Behavior Generator and SmartBody Animation

Stacy Marsella  
Research Scientist  
and SmartBody Project Leader

Andrew N Marshall  
Lead SmartBody Developer

Jina Lee  
NVBG Developer  
and PhD Candidate

# Talk Outline

- SAIBA Agent Framework (10 min)
- Motivating Example (10 min)
- BML and SmartBody (60 min)
- Break (15 min)
- From FML to BML using NVBG (50 min)
- Example Revisited (30 min)
- Beyond the Tutorial with SmartBody (5 min)

# Take Away

- Understanding of the modules and how they interact with each other.
- Ability to control the character's behavior using SmartBody and BML.
- Ability to edit and write NVBG rules that contextually generate BML behaviors for speech.

# SAIBA

<b>S</b> ituation	– World state, Perception
<b>A</b> gent	– Goals, Emotion, Planning
<b>I</b> ntention	– What to express
<b>B</b> ehavior	– How to express
<b>A</b> nimation	– Realized expression

- Message-based interface with standardized content for swappable components.
- Agent modeled as a bi-directional pipeline.
  - Feedback may incur replanning upstream.
- Facilitate reuse and sharing. Allow researchers and developers to specialize.

# SAIBA Intention

- Functional Markup Language (FML)
  - “Communicative” Function
- Examples
  - Affect
  - Grounding
  - Turn Taking
  - Intentionality
- Heylen et al, The Next Step Towards a Function Markup Language, IVA 2008

# SAIBA Behavior

- Behavior Markup Language (BML)
- Abstract description of actions
- Examples
  - Hand Gestures
  - Facial Expressions
  - Speech / Verbal Expression
  - Locomotion and Body Orientation

# VHuman's Framework

- Behavior Planning is split in two:
  - Speech behavior's surface text is provided by NLG or a dialogue database like NPC Editor.
  - NVBG infers some intention from the surface text.
- Behavior Realization is not just a conceptual module, but a separate process.
  - BML inputs via message server
  - Skeletal animation over TCP & UDP “BoneBus”
- SmartBody behaviors rely on asset references.

# Example

Surface Text: “I was mad at him.”

- Identify communicative functions:
  - Rheme / focus is “mad”
  - Lesser emphasis on “I” and “him”
  - Negative affect: Disapproval



# Example

Surface Text: “I was mad at him.”

- Possible behaviors:
  - Strong beat (gesture and/or nod) on “mad”
  - Express disapproval with facial expressions
  - Referential / deictic cues of “I” and “him”
  - Gaze at referent, addressee, or even offset, expressing some emotion

# Example

Surface Text: “I was mad at him.”

- Filter Behaviors...
  - Strong beat (gesture and/or nod) on “mad”
  - Express disapproval with facial expressions
  - Referential / deictic cues of “I” and “him”
  - Gaze at referent, addressee, or even offset, expressing some emotion

# Example

Surface Text: "I was mad at him."

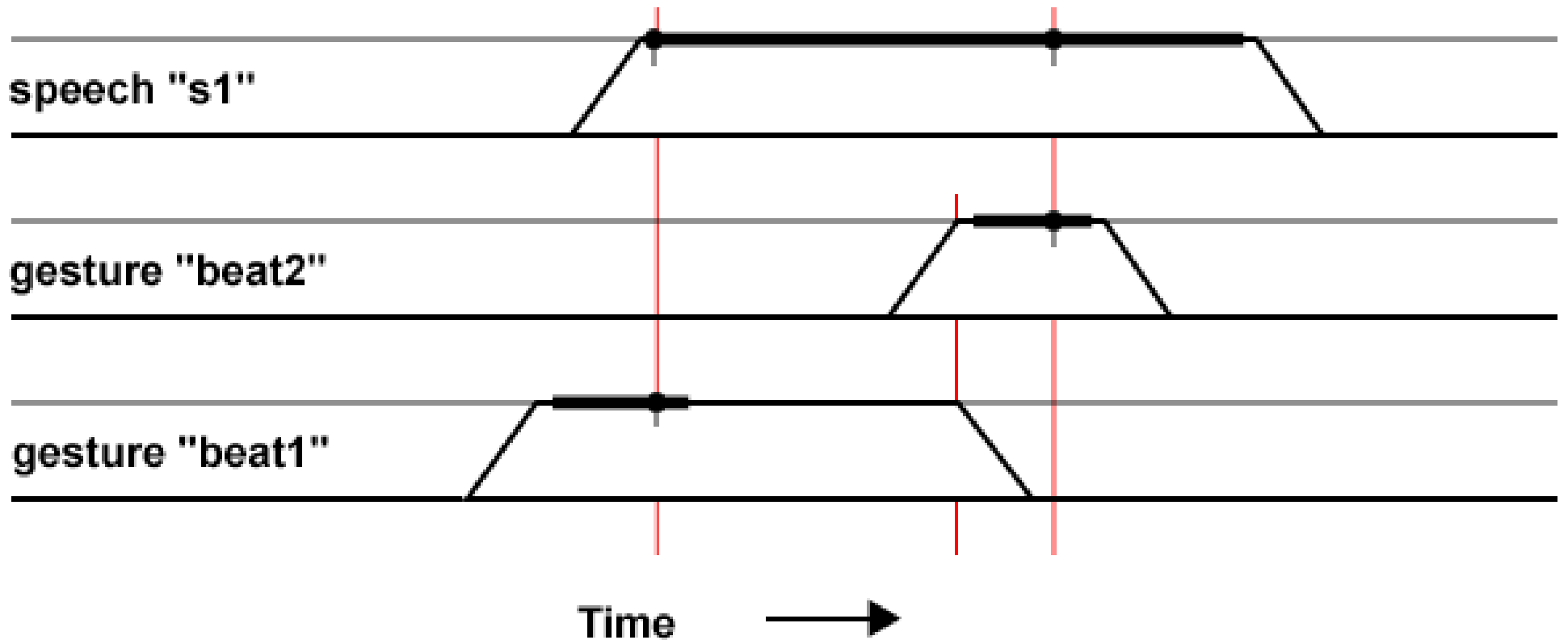
BML Request (simplified here):

```
<bml>  
  <speech>I was mad at him</speech>  
  <!-- beat gesture -->  
  <head type="nod"/>  
  <animation name="HandsAtSide_RArm_FistsChop"/>  
  <!-- facial expression -->  
  <face type="fac" au="4"/><!-- brow lowerer -->  
  <!-- gaze offset -->  
  <gaze direction="DOWNLEFT" angle="30" />  
</bml>
```

# Example

Surface Text: "I was mad at him."

Synthesize Speech & Schedule Animation:



# Example

Surface Text: “I was mad at him.”

Invoke Animation & Animation:



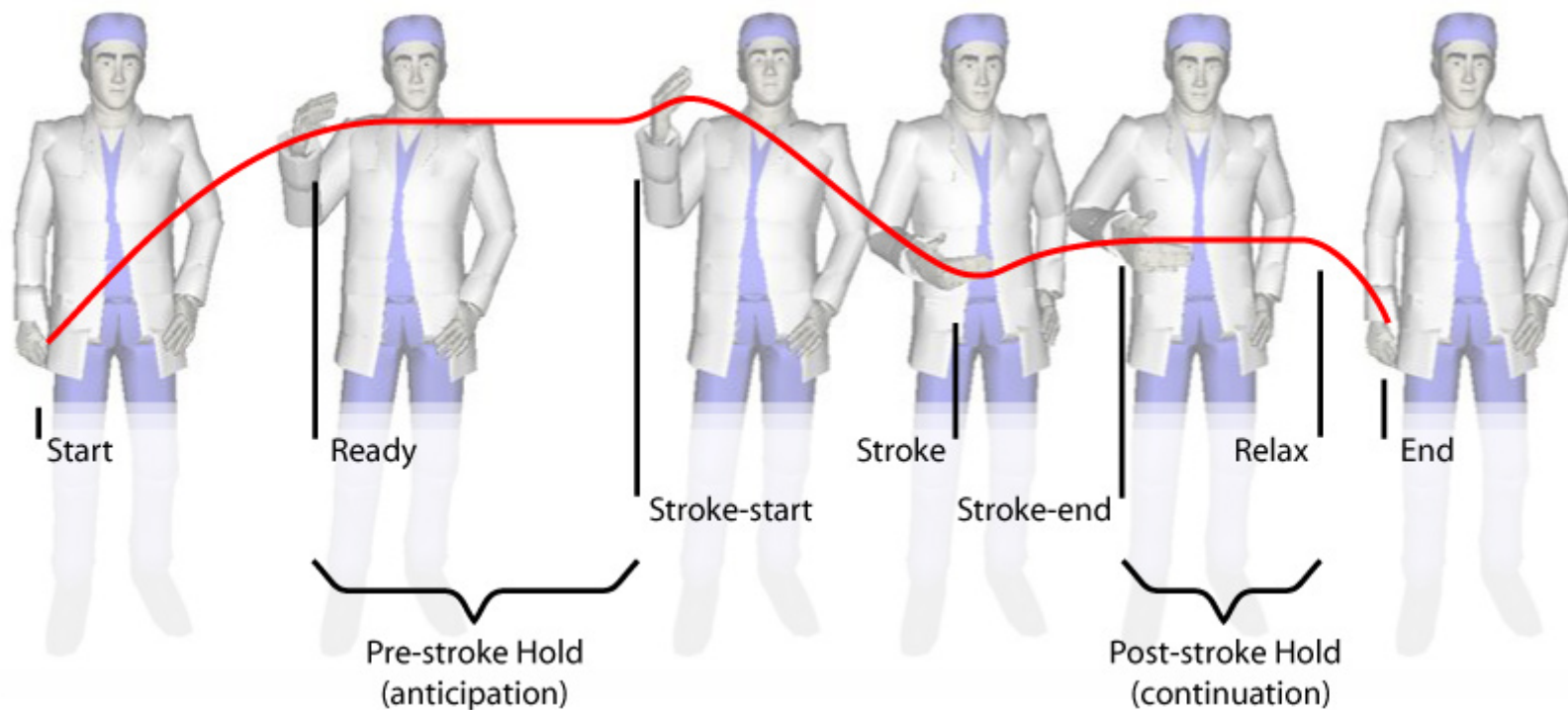
# Behavior Markup Language

- Each request is a set of behaviors and a set of constraints.
- Behaviors
  - Described independent of realization
  - No asset refs: skeleton joints, animation, files, etc.
  - General purpose and designed to overlap.
- Constraints
  - Scheduling and Timing Coordination
  - Required Behaviors

# Core Behaviors

- <face> for eye and mouth expression and motion
- <gaze> for eye direction
- <gesture> for coordinated arm and hand movement
- <head> for orientation and rhythmic nods/shakes
- <lips> for visemes
- <locomotion> for moving around an environment
- <posture> for general body configuration
- <speech> for verbal behavior
- <wait> for pauses and external coordination

# Synchronization Constraints





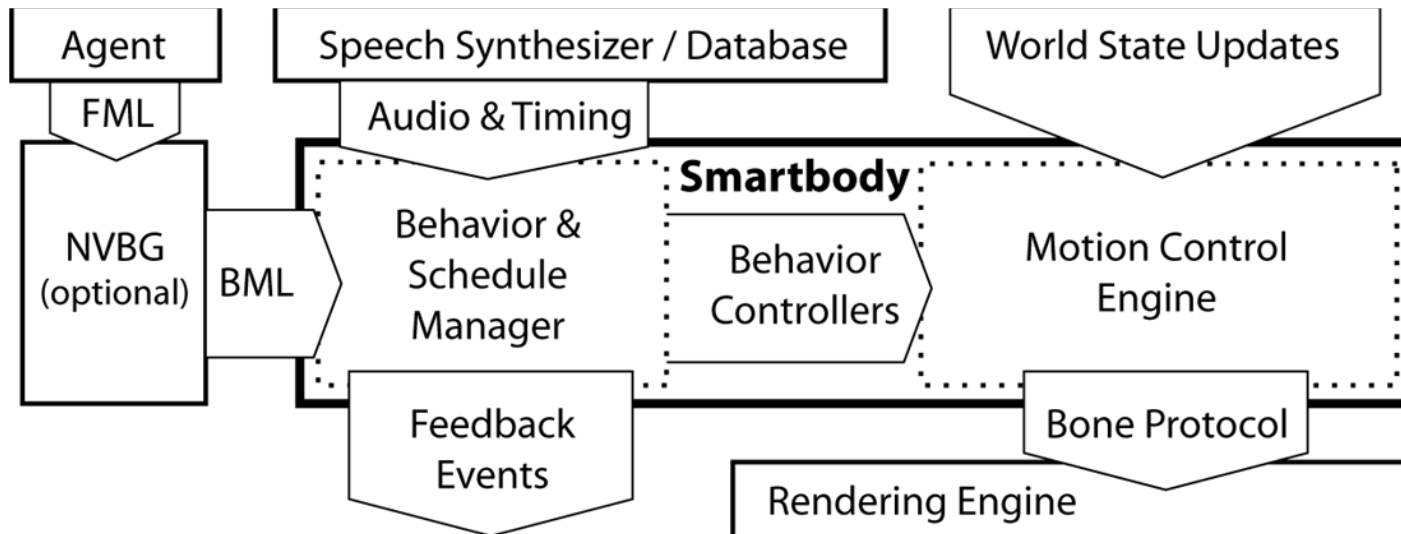
# BML References

- BML Community:
  - Forum  
<http://forum.mindmakers.org/index.php?board=20>
  - Wiki Specification Draft  
<http://wiki.mindmakers.org/projects:BML:main>
- Papers
  - Kopp et al, Towards a Common Framework for Multimodal Generation: The Behavior Markup Language, IVA 2006
  - Vilhjalmsson et al, The Behavior Markup Language: Recent Developments and Challenges, IVA 2007

# SmartBody

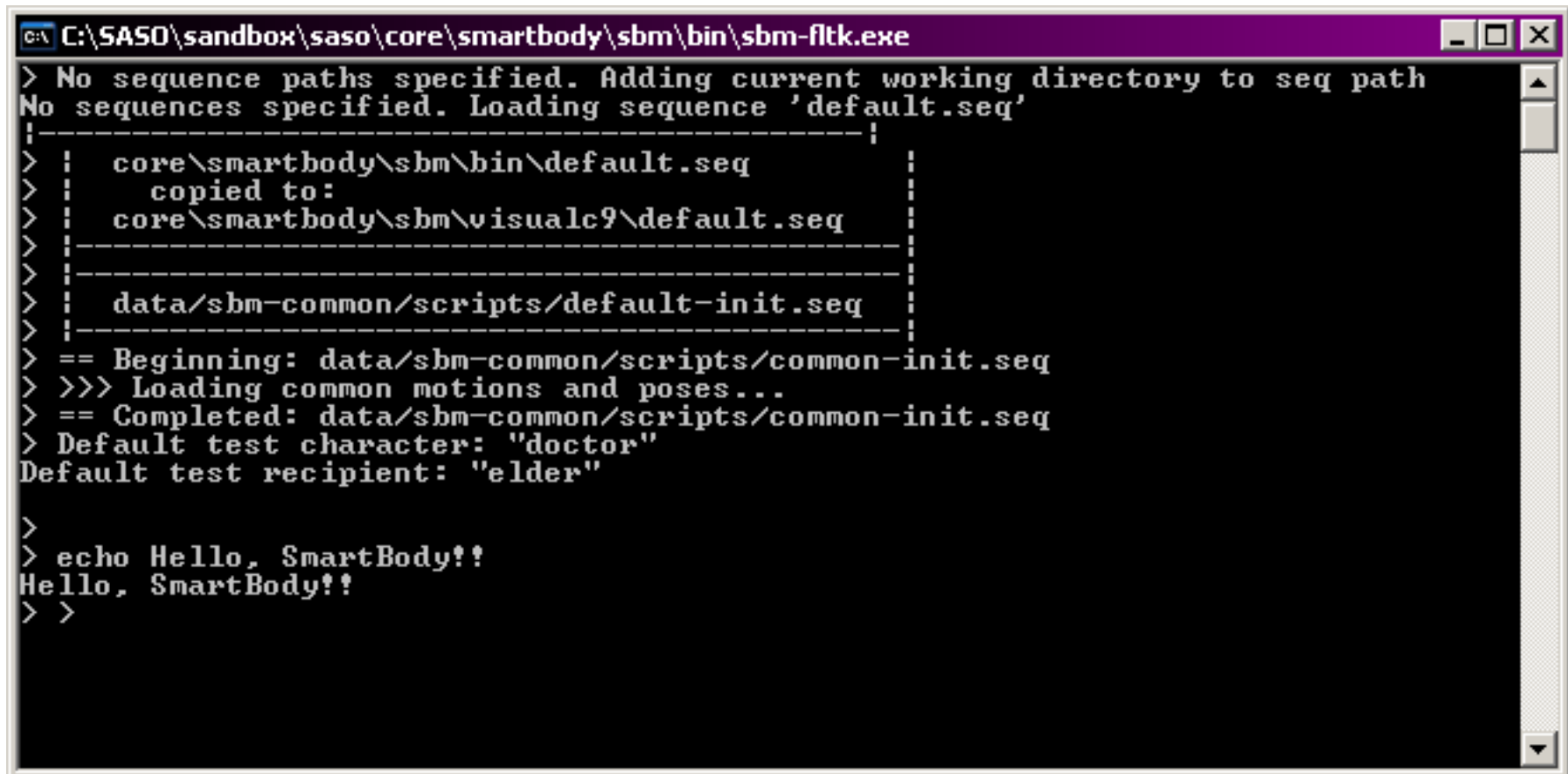
- USC's BML Implementation
- Open Source
  - <http://www.smartbody-anim.org/>
- SBM is the “SmartBody Module”
  - VHuman executable for the SmartBody library

# SBM Architecture



# Launching SBM Standalone

- sbm-flt.exe or sbmd-flt.exe (debug version)



```
C:\SASO\sandbox\saso\core\smartbody\sbm\bin\sbm-flt.exe
> No sequence paths specified. Adding current working directory to seq path
No sequences specified. Loading sequence 'default.seq'
-----|
> core\smartbody\sbm\bin\default.seq
> copied to:
> core\smartbody\sbm\visualc9\default.seq
> -----|
> data/sbm-common/scripts/default-init.seq
> -----|
> == Beginning: data/sbm-common/scripts/common-init.seq
> >>> Loading common motions and poses...
> == Completed: data/sbm-common/scripts/common-init.seq
> Default test character: "doctor"
Default test recipient: "elder"
>
> echo Hello, SmartBody??
Hello, SmartBody??
> >
```

The SBM console window

# Launching SBM Standalone

- DOS window is a log and a command prompt
  - Prints initialization info at the top
  - Searches for “default.seq” if no sequence is specified (two characters and the view window)
  - Example commands (more later):
    - “echo Hello World”
    - “test bml head nod”
    - “quit”

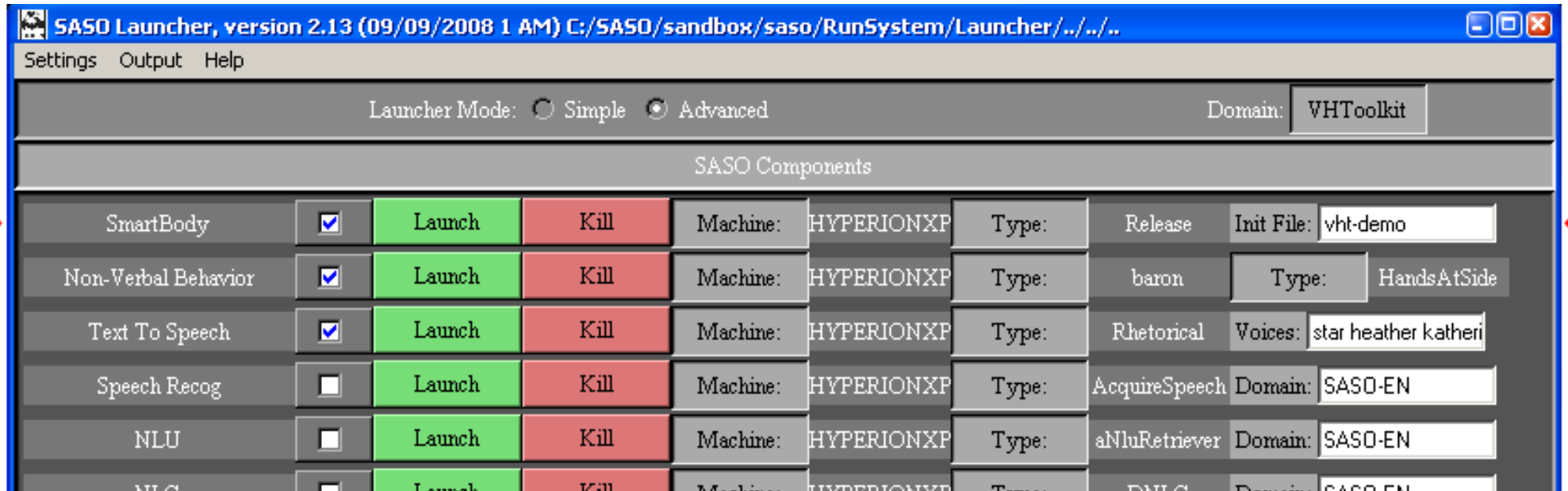
# Sequence Files

- “default.seq” and “vht-demo.seq” are initialization “sequence files”
- Sequences are text file schedules of commands
- Simple syntax:
  - Each line is a command, comment (# prefix), or blank
  - Commands are prefixed with a number declaring time in seconds after the sequence begins.
  - Rest of line is command, just like the DOS window

# SBM Commands

- Notable commands:
  - test bml character baron <head type="NOD" />  
Broadcasts BML nod request message to baron
  - test bml help  
Prints out additional BML test commands
  - seq my-sequence  
Invokes "my-sequence.seq" from the seq path
- Remote commands via "sbm" message header
- SBM Command Documentation:  
<http://smartbody.wiki.sourceforge.net/SBM+Commands>

# Launching SBM for VHuman



The screenshot shows the SASO Launcher application window. The title bar reads "SASO Launcher, version 2.13 (09/09/2008 1 AM) C:/SASO/sandbox/saso/RunSystem/Launcher/../../..". The menu bar includes "Settings", "Output", and "Help". The "Launcher Mode" is set to "Advanced" (selected with a radio button), and the "Domain" is "VHToolkit".

The main area is titled "SASO Components" and contains a table of components. Each component has a checkbox for selection, a "Launch" button (green), a "Kill" button (red), and various configuration fields. Two red arrows point to the left and right sides of the component table.

Component	Selected	Launch	Kill	Machine	Type	Release	Init File
SmartBody	<input checked="" type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	Release	vht-demo
Non-Verbal Behavior	<input checked="" type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	baron	HandsAtSide
Text To Speech	<input checked="" type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	Rhetorical	Voices: star heather katheri
Speech Recog	<input type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	AcquireSpeech	Domain: SASO-EN
NLU	<input type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	aNluRetriever	Domain: SASO-EN
NLC	<input type="checkbox"/>	Launch	Kill	HYPERIONXP	Type:	DNLC	Domain: SASO-EN



# SBM Configuration

- Launcher configures command line
  - -seqpath, -mepath, -seq, and -host
  - <http://smartbody.wiki.sourceforge.net/SBM+Command+Line>
- “saso/data/sbm-saso/scripts/vht-demo.seq” configures...
  - Asset paths (skeletons, animations, poses)
  - Characters
  - Pawns (named locations)
  - Renderer camera

# Sending BML to SBM

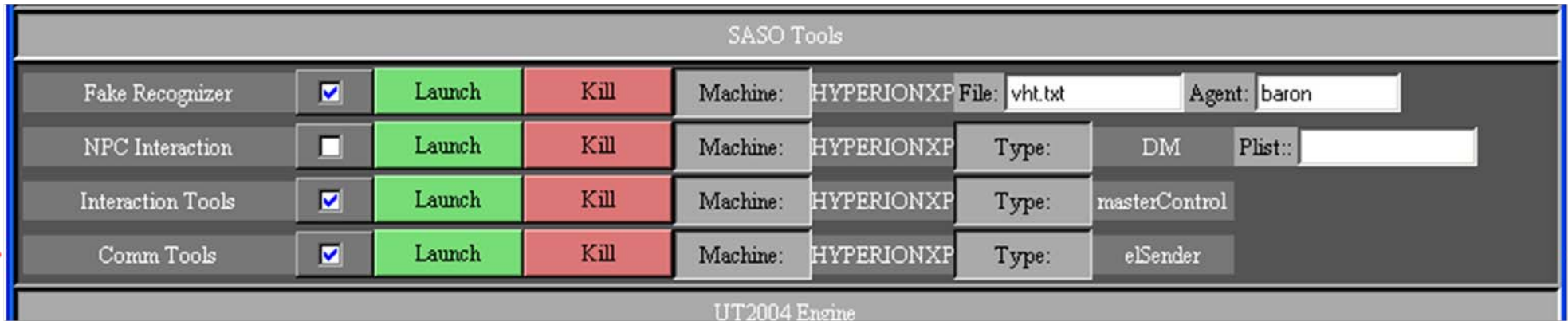
- vrSpeak Message Format

- actor, recipient, message-id, XML

```
vrSpeak baron detective msg-123
<?xml version="1.0" encoding="UTF-8"?>
<act>
  <bml>
    <speech>Hello</speech>
  </bml>
</act>
```

- <http://smartbody.wiki.sourceforge.net/VHuman+BML+Messages>
- <http://smartbody.wiki.sourceforge.net/SmartBody+BML>

# Remote Testing with elSender



- elSender is a simple message sender and logger
- Launch elSender, under Comm Tools
- In the elSender window, disable auto-complete
- Send the message (with SBM, Unreal running)

```
sbm test bml <head type="NOD" />
```

# Remote Testing with BML Files

- Open the file

```
saso/data/sbm-saso/ExampleBML/test-yes.xml
```

- Save it a new filename, such as

```
saso/data/sbm-saso/ExampleBML/test-vht.xml
```

- In eISender, execute the command

```
sbm test bml file../../../../data/sbm-saso/ExampleBML/test-vht.xml
```

- Now we can edit the BML and test iterations by resending the same command in eISender

# Behaviors from the Example

- Speech: “I was mad at him.”
- Gestures
  - Beats, Deictics
- Head nod beat
- Facial expressions for disapproval
- Gaze at referent, addressee, or even offset, expressing some emotion

# Speech

- Basic Form:

```
<speech>I was mad at him.</speech>
```

- Attribute “type” is assumed to be “plain/text”.

- Type can also be SSML

```
<speech type="application/ssml+xml">  
  I was <emphasis>mad</emphasis> at him.  
</speech>
```

- One one speech behavior per request, and it must be the first behavior.

- <http://smartbody.wiki.sourceforge.net/BML+Speech+Element>

# Animated Gestures

- Basic Form:

```
<sbm:animation  
    name="HandsAtSide_RArm_MidBeat" />
```

- Value of “name” is the animation filename base.
- Animation files (.skm's) are located
  - saso/data/sbm-common/common-sk/motion
  - saso/data/sbm-saso/doctor/motion
- Prefer animations with matching pose:
  - HandsAtSide\_\*
- <http://smartbody.wiki.sourceforge.net/BML+Animation+Element>

# Body Posture

- Basic Form:

```
<body posture="HandsAtSide" />
```

- “posture” may be pose (.skp) or looped motion (.skm):

- saso/data/sbm-common/common-sk/pose
- saso/data/sbm-common/common-sk/motion
- saso/data/sbm-saso/doctor/pose
- saso/data/sbm-saso/doctor/motion

- Example motion texture from motions directory:

```
<body posture="HandsAtSide_Motex" />
```

- <http://smartbody.wiki.sourceforge.net/BML+Body+Element>



# Posture Transitions

- Combine posture and motion together:

```
<body posture="CrossedArms_Motex" />  
<sbm:animation  
  name="HandsAtSide_Transition_CrossedArms"  
>
```

- Other postures with **\*\_Transition\_\*** files:
  - LHandOnHip
  - HandsOnHip

# Head Nods and Shakes

- Basic Forms:

```
<head type="NOD" />
```

```
<head type="SHAKE" />
```

- Optional attributes with defaults:

- `repeats="1.0"` (one full sine cycle)

- `amount="0.5"` (half of full extension, maybe negative to invert)

- `velocity="1.0"` (one cycle per second)

- <http://smartbody.wiki.sourceforge.net/BML+Head+Element>

# Facial Expressions

- Basic Form:

```
<face type="FACS" au="4" />
```

- Optional Attributes with Defaults

- amount="0.5" (amplitude of action unit)

- <http://smartbody.wiki.sourceforge.net/BML+Face+Element>  
including list of implemented action units

- FAC Action Unit Examples:

- <http://www.cs.cmu.edu/afs/cs/project/face/www/facs.htm>

# Gaze: Target and Offset

- Basic Form:

```
<gaze target="detective" />
```

- Offset Attributes:

```
<gaze target="detective"  
      direction="UPLEFT" angle="45"  
/>
```

```
<gaze target="detective"  
      direction="POLAR 20" angle="45"  
/>
```

<http://smartbody.wiki.sourceforge.net/BML+Gaze+Element>

# Gaze: Abstract Spine

- Our gaze controller abstracts the spine and eyes into five control keys:

- Eyes, Head, Neck, Chest, Back

- Gaze can be limited to a subset of keys:

```
<gaze target="detective"  
      sbm:joint-range="NECK EYES"  
>
```

- Unaddressed keys may still be controlled by prior gaze commands

<http://smartbody.wiki.sourceforge.net/BML+Gaze+Element>

# Gaze: Speed and Smoothing

- Optional attributes with defaults:

```
<gaze target="detective"  
      sbm:joint-speed="1000 1500 2000"  
      sbm:speed-smoothing="0.8 0.8 0.1"  
>
```

- Defines the speed/smoothing for lower back (lumbar), neck, and eyes, in order.

<http://smartbody.wiki.sourceforge.net/BML+Gaze+Element>

# Gaze Demo

# Putting it all together...

```
<bml>  
  <speech>I was mad at him.</speech>  
  <!-- beat gesture -->  
  <head type="nod" />  
  <animation name="HandsAtSide_RArm_FistsChop" />  
  <!-- facial expression -->  
  <face type="facs" au="4" /><!-- brow lowerer -->  
  <!-- gaze offset -->  
  <gaze direction="DOWNLEFT" angle="30" />  
</bml>
```

•Except...



**Synchronization!!!**

# SmartBody Sync-Points

- SmartBody supports start, ready, stroke, and relax, but not stroke\_start and stroke\_end.
- Uses standard BML attribute notation:

```
<sbm:animation
  id="gest1"
  stroke="0" stroke="HandsAtSide_RArm_MidBeat"
/> name="HandsAtSide_RArm_MidBeat"
/> <head type="NOD"
  stroke="gest1:stroke"
/> stroke="gest1:stroke"
/>
```

# Speech Synchronization

```
<speech type="text/plain" id="sp">  
  I was <tm id="mad"/>mad at him.  
</speech>  
<sbm:animation  
  name="HandsAtSide_RArm_MidBeat"  
  stroke="sp:mad"  
>
```

## ***Or, if using SSML...***

```
<speech type="application/ssml+xml" id="sp">  
  I was <mark name="mad"/>mad at him.  
</speech>  
<sbm:animation  
  name="HandsAtSide_RArm_MidBeat"  
  stroke="sp:mad"  
>
```

# Example, Synchronized

- Add an id element to speech:

```
id="sp1"
```

- Insert `<tm id=".." />` tags at appropriate word breaks:

- Before “I”, “mad”, and “him”
- After all words

- For beat, align the stroke of nod and gesture with “mad”:

```
stroke="sp1:mad"
```

# Blending Behaviors

- Animation controllers are blended in a specific order:
  - Location, Posture, Animation, Gaze, Nod, Face
- Some controllers interact, like Gaze and Nod
- Normally, this is what you want, but sometimes one controller can mask another
  - E.g., Gaze can mask an predominantly spinal animation

# Misc. Notes

- Currently, there is no way to cancel a gaze. Instead, just define a new target to gaze with a new gaze behavior:

```
<gaze target="detective" />
```

# SmartBody References

- SmartBody wiki:
  - “Manual” link from <http://www.smartbody-anim.org/>
  - Reference section at the bottom of the Table of Contents.
    - “VHuman Messages” are primary interfaces
    - “SBM Commands” are for the interactive prompt
    - “SmartBody BML” details the implemented BML behaviors

# From FML to BML using NVBG

Presented by Jina



# SmartBody

## The Open Source Project

- Website
  - <http://www.smartbody-anim.org/>
- Mailing Lists:
  - <http://smartbody.wiki.sourceforge.net/Contact>
- Repository
  - Via SVN. Same VHuman Layout.
- Renderers:
  - Internal (Solid body, OpenGL)
  - Ogre
  - Soon Panda 3D, thanks to Reykjavik University

# SmartBody

## Getting Involved

- Contact:
  - <http://smartbody.wiki.sourceforge.net/Contact>
- Roadmap:
  - <http://smartbody.wiki.sourceforge.net/Roadmap>
- Adding new / improving rendering engines:
  - <http://smartbody.wiki.sourceforge.net/BoneBus>
- Adding new / improving BML behaviors:
  - <http://smartbody.wiki.sourceforge.net/Extending+SmartBody+BML>
- Adding new / improving controllers:
  - <http://smartbody.wiki.sourceforge.net/Creating+New+Animation+Controllers>