JavaDoc Documentation of the MIM Package

Boi Sletterink
boi@linuxfan.com

August 9, 1999

Contents

Chapter 1

EDU.rutgers.stimulate.MIM.SpeechParsing

Package Contents *Page*

1.1 Classes

1.1.1 *Class* Alternative

Parses and contains info about a recognized alternative. There is probably a major bug in here concerning timestamps that start the beginning of the current day (0:00) in seconds vs. currentTimeMillis(). We need to look into that, however, currently the timestamps are not used anyway.

Declaration

public class Alternative
extends java.lang.Object

Fields

- public static final int maxWords
  –
- public int numWords
  –
- public String words
  –
- public int scores
  –
- public long starts
  –
- public long ends
  –

- public int score
  –
- public long timeBegin
  –
- public long timeEnd
  –
- public boolean valid
  –

Constructors

- *Alternative*
  `public Alternative( )`
- *Alternative*
  `public Alternative( java.lang.String` **sentence**`, long` **timeStamp** `)`
- *Alternative*
  `public Alternative( java.io.StreamTokenizer` **tokens** `)`

Methods

- *parseTime*
  `public static int parseTime( java.io.StreamTokenizer` **tokens** `)`
- *parseFormattedTime*
  `public static int parseFormattedTime( java.io.StreamTokenizer` **tokens** `)`
- *toString*
  `public String toString( int` **base** `)`
- *toString*
  `public String toString( )`
- *match*
  `public boolean match( java.lang.String s, int` **position** `)`
  – Usage
    ∗ Checks if the string matches the alternative, starting from the given word.
  – Parameters
    ∗ `s` - the string to be matched.
    ∗ `position` - the starting word in the alternative.
  – Returns - boolean true if there is a word-by-word match, otherwise false.
- *breakString*
  `public static String breakString( java.lang.String s )`
  – Usage
    ∗ Breaks a string into words. The implementation is kinda lazy in the sense that the string should not begin nor end with a space. It *will* break for those strings.

      The definition of a word is any sequence of non-space (' ') characters. Newlines, carriage returns, punctuation etc. will all be considered as a normal character that does not separate words.
  – Parameters
    ∗ `s` - the string to be broken into words.
  – Returns - String[] an array of words in the same sequence as the appear in the string. The array is as long as the number of words.

1.1.2 *Class* AlternativeClassifier

Classifies incoming alternatives.

Declaration

public class AlternativeClassifier
extends java.lang.Object

Fields

- public Classifier classes
  –

Constructors

- *AlternativeClassifier*
  `public AlternativeClassifier( java.lang.String` **classifiles** `)`

Methods

- *alternativeAcceptable*
  `public boolean alternativeAcceptable(`
  `EDU.rutgers.stimulate.MIM.SpeechParsing.Alternative` **alt** `)`

1.1.3 *Class* Classifier

Classifies incoming alternatives.

Declaration

public class Classifier
extends java.lang.Object

Fields

- public static double BoundaryScaleParameter
  –
- public int classDefinition
  –
- public double maxDistance
  –

Constructors

- *Classifier*
  `public Classifier( java.io.Reader` **reader**`, int` **dimensions** `)`

Methods

- *isInClass*
  `public boolean isInClass( int [] vector )`
- *minDistance*
  `public static double minDistance( int [] row, int [][] signal, int [] index )`
- *norm*
  `public static double norm( int [] row, int [] signal )`

1.1.4 *Class* EmptyAlt

Parses and contains info about an empty alternative.

Declaration

public class EmptyAlt
extends EDU.rutgers.stimulate.MIM.SpeechParsing.Alternative

Constructors

- *EmptyAlt*
  `public EmptyAlt( java.io.StreamTokenizer tokens )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.SpeechParsing.Alternative`

(in 1.1.1, page 2)

- `public static int parseTime( java.io.StreamTokenizer tokens )`
- `public static int parseFormattedTime( java.io.StreamTokenizer tokens )`
- `public String toString( int base )`
- `public String toString( )`
- `public boolean match( java.lang.String s, int position )`
- `public static String breakString( java.lang.String s )`

1.1.5 *Class* Sentence

Parses and contains info about a recognized sentence.

Declaration

public class Sentence
extends java.lang.Object

Fields

- public final int maxAlternatives
  –
- public int numAlternatives
  –

- public Alternative alternatives
  –
- public boolean valid
  –

Constructors

- *Sentence*
  `public Sentence( )`
- *Sentence*
  `public Sentence( java.lang.String` **sentence**`, long` **timeStamp** `)`
- *Sentence*
  `public Sentence( java.io.StreamTokenizer` **tokens** `)`

Methods

- *altNumberTag*
  `public String altNumberTag( int` **num** `)`
- *setupTokenizer*
  `public static StreamTokenizer setupTokenizer( java.io.StreamTokenizer` **tokenizer** `)`

1.1.6 *Class* SyntaxError

SyntaxError exception class, used to raise syntax error exceptions.

Declaration

public class SyntaxError
extends java.lang.Throwable

Serializable Fields

- public String fileName
  –

Fields

- public String fileName
  –

Constructors

- *SyntaxError*
  `public SyntaxError( java.lang.String s )`

Methods

- *MakeSEMessage*
  `public static String MakeSEMessage( java.io.StreamTokenizer` **tokens**`, int` **expectedType**`, java.lang.String` **expectedValueString** `)`

Methods inherited from class `java.lang.Throwable`

- `public native Throwable fillInStackTrace( )`
- `public String getLocalizedMessage( )`
- `public String getMessage( )`
- `public void printStackTrace( )`
- `public void printStackTrace( java.io.PrintStream )`
- `public void printStackTrace( java.io.PrintWriter )`
- `public String toString( )`

Chapter 2

EDU.rutgers.stimulate.MIM

Package Contents — *Page*

Interfaces

Classes

2.1 Interfaces

2.1.1 *Interface* MI

Interface for a generic Modal Input object. Defines functions to discover about it's capabilities.

Declaration

```
public abstract interface MI
```

Fields

- public static final int POINTER_2D
  –
- public static final int POINTER_2D_DIRECTIONAL
  –
- public static final int POINTER_3D
  –
- public static final int POINTER_3D_DIRECTIONAL
  –
- public static final int NL_INPUT
  –
- public static final int NL_FEEDBACK
  –
- public static final int FORCE_FEEDBACK
  –
- public static final int FINGER_FLEXING
  –

Methods

- *getModalInputName*
  `public String getModalInputName( )`
  - Usage
    - get the name of this modal input.
  - Returns - String name.
- *getModalInputVersion*
  `public String getModalInputVersion( )`
  - Usage
    - get the version info of this modal input.
  - Returns - String version information.
- *getCapabilities*
  `public BitSet getCapabilities( )`
  - Usage
    - gets the capabilities of this input. Use the constants below.
  - Returns - BitSet the set of capabilities.
- *getControlPanel*
  `public JPanel getControlPanel( )`
  - Usage
    - Construct a control panel for use in a window, dialog or other panel.
  - Returns - JPanel the panel with the controls.
- *loadCustomizer*
  `public void loadCustomizer( int` **slot**`, java.lang.Object` **customizerData** `)`

- Usage
  - Loads a customizer into specified storage slot. Overwrites existing customizer in that slot if necessary.
- Parameters
  - `slot` - int denoting the slot number.
  - `customizerData` - Object containing customization data.

- *selectCustomizer*
  `public void selectCustomizer( int slot )`
  - Usage
    - Switches customized behaviour to the description stored in specified slot.
  - Parameters
    - `slot` - int denoting the slot number.

2.1.2 *Interface* MIApp

Defines an interface for addressing modal input apps.

Declaration

public abstract interface MIApp

Methods

- *notifyManagerStarted*
  `public void notifyManagerStarted( EDU.rutgers.stimulate.MIM.MIManager manager )`
  - Usage
    - Called by MIAdvertiser to notify a app of the start of a registered manager. Should in general not be called by a app itself; actions needed for subscription, unsubscription, registration and unregistration are to be performed by the MIMAdvertiser.

      This function is also called when there is already a manager and the app just signed up.
  - Parameters
    - `manager` - the MIManager object that just registered.
- *notifyManagerStopped*
  `public void notifyManagerStopped( EDU.rutgers.stimulate.MIM.MIManager manager )`
  - Usage
    - Called by MIAdvertiser to notify a app of the stop of a registered manager. Should in general not be called by a app itself; actions needed for subscription, unsubscription, registration and unregistration are to be performed by the MIMAdvertiser.

      This function is also called when a manager does not stop but the app just resigned.
  - Parameters
    - `manager` - the MIManager object that is going to quit.
- *getTopLevelComponent*
  `public JComponent getTopLevelComponent( )`
  - Usage
    - Called by a MIManager to findout what the toplevel component associated with this app is. This is used to determine which component is the topmost (selected) component in the desktop.

  - Returns - JComponent the component associated with this MI app.

- *notifyMIAvailable*
  `public void notifyMIAvailable( EDU.rutgers.stimulate.MIM.MI` **input** `)`
  - Usage
    * Called by MIManager to tell the app that a certain modal input is now available. This may be when the app registers with the manager or when a new input becomes available.
  - Parameters
    * `input` - the MI object providing the new input. This can be used to register a hook. Do not call any functions in this object other than to figure out what it is.

2.1.3 *Interface* MIEnhancer

Defines an interface for drop-in enhancers for non-MI apps.

Declaration

public abstract interface MIEnhancer

Methods

- *hookupWith*
  `public boolean hookupWith( java.awt.Component` **bean** `)`
  - Usage
    * The empty constructor. An MIEnhancer must specify an empty constructor. This is to support the Class.newInstance() method, which is used to create the new instance.
      public MIEnhancer();

      /** Called by the MIManager to hook this MIEnhancer to it's application bean. This is the place where the enhancer registers it's MIApp object with the manager, if it wants to do so.
  - Parameters
    * `bean` - the application bean to be enhanced.

2.1.4 *Interface* MIManager

Defines an interface for addressing modal input managers.

Declaration

public abstract interface MIManager

Methods

- *getMIManagerName*
  `public String getMIManagerName( )`
  - Usage
    * Return the name of this manager.
  - Returns - String name of the manager.

- *getMIManagerVersion*
  `public String getMIManagerVersion( )`

  - Usage
    - Returns version information.
  - Returns - String version information.

- *getModalInputs*
  `public Enumeration getModalInputs( )`
  - Usage
    - Lists all modal input opbjects.
  - Returns - Enumeration of all modal input objects.

- *getMIApps*
  `public Enumeration getMIApps( )`
  - Usage
    - Lists all registered MIApp objects.
  - Returns - Enumeration of all registered multi-modal aware objects.

- *isRegisteredApp*
  `public boolean isRegisteredApp( java.lang.Object` **app** `)`
  - Usage
    - Checks if the object is a registered MIApp object. The provided object does not have to be an MIApp object, so other objects can just check any object for being registered. If it's not an MIApp object, isRegistered() will just always return false.
  - Parameters
    - `app` - the object to check.
  - Returns - boolean true if the object is registered.

- *getAppForComponent*
  `public MIApp getAppForComponent( javax.swing.JComponent` **component** `)`
  - Usage
    - Checks if the component is the toplevel component of a registered app and returns the corresponding app object.
  - Parameters
    - `component` - the JComponent that is to be checked.
  - Returns - MIApp the rgistered app, if any. Otherwise null.

- *addRawMIPointer2DListener*
  `public void addRawMIPointer2DListener( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MIMPointer2DEventListener` **target**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input** `)`
  - Usage
    - Register for events from this specific modal input. This will have the MIM send the object all 2D pointer events from the specified modal input while it's outer component is topmost.

      Until further notice, app and target have to be the same.
  - Parameters
    - `app` - the app associated with the target.
    - `target` - the target for the events.
    - `input` - the modal input to listen to.

- *removeRawPointer2DListener*
  `public void removeRawPointer2DListener(`
  `EDU.rutgers.stimulate.MIM.MIMPointer2DEventListener` **target**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input** `)`
  - Usage
    - Unregister for 2D events from a certain input.

- Parameters
  * `target` - the target listener.
  * `input` - the target input.

- *addNLIStringListener*
  `public void addNLIStringListener( EDU.rutgers.stimulate.MIM.NLIStringListener` **target** `)`
  - Usage
    * Register for NLI events in String format.
  - Parameters
    * `target` - the target for the input.

- *removeNLIStringListener*
  `public void removeNLIStringListener( EDU.rutgers.stimulate.MIM.NLIStringListener` **target** `)`
  - Usage
    * Unregister for NLI events in String format.
  - Parameters
    * `target` - a registered target.

- *addNLISentenceListener*
  `public void addNLISentenceListener( EDU.rutgers.stimulate.MIM.NLISentenceListener` **target** `)`
  - Usage
    * Register for NLI events in Sentence format.
  - Parameters
    * `target` - the target for the input.

- *removeNLISentenceListener*
  `public void removeNLISentenceListener( EDU.rutgers.stimulate.MIM.NLISentenceListener` **target** `)`
  - Usage
    * Unregister for NLI events in Sentence format.
  - Parameters
    * `target` - a registered target.

- *lockPointerInput*
  `public void lockPointerInput( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MI` **input**`, boolean` **lock** `)`
  - Usage
    * Locks/unlocks all pointer input to a specific modality.
  - Parameters
    * `app` - the MI app.
    * `input` - the input to (un)lock.
    * `lock` - set to true to lock.

- *setPointerMotionInput*
  `public boolean setPointerMotionInput( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input** `)`
  - Usage
    * Sets default input for MouseMoved events. Only this input will generate MouseMoved events. Other mouse events, such as clicks, will get through.
  - Parameters
    * `app` - the MI app.

    - `input` - the input to generate MouseMotion-events for.
  - Returns - boolean true when successful.

- *findEvent*
  `public MIEvent findEvent( EDU.rutgers.stimulate.MIM.MI` **input**`, long` **when** `)`
  - Usage
    - Returns an event from the history of the specified input at the specified time. If there was no event at that specific moment, it returns the nearest or null if it was too long ago.
  - Parameters
    - `input` - the input where the event should come from.
    - `when` - the moment when it should have been.
  - Returns - MIEvent the requested event, or null if the input was invalid or the moment was too long ago.
- *convertPoint*
  `public Point convertPoint( EDU.rutgers.stimulate.MIM.RawPointer2DEvent` **event**`, java.awt.Component` **c** `)`
  - Usage
    - Returns the point where an event occurred relative to the component's origin. This is used to transform from the pointer's coordinate space to the component's.
  - Parameters
    - `event` - the pointer event.
    - `c` - the component.
  - Returns - Point the point in the bean's space.
- *generateClickedEvent*
  `public boolean generateClickedEvent( EDU.rutgers.stimulate.MIM.MIApp` **app**`, EDU.rutgers.stimulate.MIM.MIPointer` **input**`, long` **when**`, int` **count** `)`
  - Usage
    - Generates a single mouseclick from a specific pointer. The app needs to specify itself as a reference, an input and moment to generate the coordinates from. generateClickEvent() will check if the requested modality was actually in scope for the app at the specified time. If not, or the manager cannot know (too long ago) it will return false.

      Note: scope checking is currently not implemented. The app variable is not used.
  - Parameters
    - `app` - the app.
    - `input` - the input to generate the click from.
    - `when` - the time when the event should have occurred.
    - `count` - the number of clicks wanted.
  - Returns - boolean true if successful.
- *generatePressedEvent*
  `public boolean generatePressedEvent( EDU.rutgers.stimulate.MIM.MIApp` **app**`, EDU.rutgers.stimulate.MIM.MIPointer` **input**`, long` **when** `)`
  - Usage
    - Generates a mouse pressed event from a specific pointer. The app needs to specify itself as a reference, an input and moment to generate the coordinates from. generatePressedEvent() will check if the requested modality was actually in scope for the app at the specified time. If not, or the manager cannot know (too long ago) it will return false.

      Note: scope checking is currently not implemented. The app variable is not used.
  - Parameters
    - `app` - the app.
    - `input` - the input to generate the press from.

    * `when` - the time when the event should have occurred.
  - Returns - boolean true if successful.

- *startDragging*
  `public boolean startDragging( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input**`, long` **when** `)`
  - Usage
    * Starts a dragging motion with a specific pointer. The app needs to specify itself as a reference, an input and moment to generate the coordinates from. startDragging() will check if the requested modality was actually in scope for the app at the specified time. If not, or the manager cannot know (too long ago) it will return false.

      startDragging() generates a MOUSE_PRESSED event and subsequent MOUSE_DRAGGED events until stopDragging() is called. All motion events of the specified pointer input will be sent as MOUSE_DRAGGED events to the component that was underneath the pointer when the dragging started.

      Note: scope checking is currently not implemented. The app variable is not used.
  - Parameters
    * `app` - the app.
    * `input` - the input to generate the press from.
    * `when` - the time when the event should have occurred.
  - Returns - boolean true if successful.

- *generateReleasedEvent*
  `public boolean generateReleasedEvent( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input**`, long` **when** `)`
  - Usage
    * Generates a mouse released event from a specific pointer. The app needs to specify itself as a reference, an input and moment to generate the coordinates from. generateReleasedEvent() will check if the requested modality was actually in scope for the app at the specified time. If not, or the manager cannot know (too long ago) it will return false.
  - Parameters
    * `app` - the app.
    * `input` - the input to generate the press from.
    * `when` - the time when the event should have occurred.
  - Returns - boolean true if successful.

- *stopDragging*
  `public boolean stopDragging( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input**`, long` **when** `)`
  - Usage
    * Stops a dragging motion with a specific pointer. The app needs to specify itself as a reference, an input and moment to generate the coordinates from. stopDragging() will check if the requested modality was actually dragging a component from the app at the specified time. If not, it will return false.

      stopDragging() generates a MOUSE_RELEASED event and stops the sending of MOUSE_DRAGGED events.

      Note: scope checking is currently not implemented. The app variable is not used.
  - Parameters
    * `app` - the app.

    * `input` - the input to generate the press from.
    * `when` - the time when the event should have occurred.
  - Returns - boolean true if successful.

- *sendNLFeedback*
  `public boolean sendNLFeedback( java.lang.String` **feedback** `)`
  - Usage
    * Outputs the specified string, which should be natural language, to the user. The manager will select one or more feedback providing modalities if there is more than one.
  - Parameters
    * `feedback` - the feedback String.
  - Returns - boolean true if successful.
- *setGrammar*
  `public void setGrammar( EDU.rutgers.stimulate.MIM.MIApp` **app**`,`
  `EDU.rutgers.stimulate.MIM.Grammar` **grammar** `)`
  - Usage
    * Set a new grammar for this application.
  - Parameters
    * `grammar` - the Grammar object.
- *registerApp*
  `public void registerApp( EDU.rutgers.stimulate.MIM.MIApp` **app** `)`
  - Usage
    * Register a MIApp object.
  - Parameters
    * `app` - the MIApp object.
- *unregisterApp*
  `public void unregisterApp( EDU.rutgers.stimulate.MIM.MIApp` **app** `)`
  - Usage
    * Unregisters an MIApp object.
  - Parameters
    * `app` - the MIApp object.
- *registerPointer2DModalInput*
  `public MIMPointer2DEventListener registerPointer2DModalInput(`
  `EDU.rutgers.stimulate.MIM.MIPointer` **input** `)`
  - Usage
    * Register a 2D pointer input object.
  - Parameters
    * `input` - the MI object.
  - Returns - MIPointer2DEventListener the adapter to send things to. May be null if the manager refuses to receive input from it. This might be the case when a second input of the same kind is registered.
- *registerNLIModalInput*
  `public MIMNLIEventListener registerNLIModalInput( EDU.rutgers.stimulate.MIM.MI` **input**
  `)`
  - Usage
    * Register a natural language input object.
  - Parameters
    * `input` - the MI object.
  - Returns - MIEventListener the adapter to send things to. May be null if the manager refuses to receive input from it. This might be the case when a second input of the same kind is registered.

- *unregisterModalInput*
  `public void unregisterModalInput( EDU.rutgers.stimulate.MIM.MI` **input** `)`
  - Usage
    * Unregisters a modal input.
  - Parameters
    * `input` - the input object.
- *registerEnhancer*
  `public void registerEnhancer( java.lang.Class` **listenTo**`, java.lang.Class` **enhancer** `)`
  - Usage
    * Registers a MIEnhancer. This will cause the manager to create an instance of the requested enhancer class whenever an instance of the bean class enters it's workspace. It is possible to register multiple times for the same class to listen to (eg. JComponent). However, it is not possible to register the same enhancer twice. If your enhancer needs to listen for two or more classes, listen to JComponent and only attach to the new component when it is one you support.
  - Parameters
    * `listenTo` - the supported class. This is the class the manager will look for.
    * `enhancer` - the accompanying enhancer class. This must be a class object for a class implementing MIEnhancer.
- *unregisterEnhancer*
  `public void unregisterEnhancer( java.lang.Class` **enhancer** `)`
  - Usage
    * Unregisters an MIEnhancer.
  - Parameters
    * `enhancer` - stop looking for instances of this class.

2.1.5 *Interface* MIMNLIEventListener

Event listener interface for raw Natural Language Interface events.

Declaration

public abstract interface MIMNLIEventListener

Methods

- *IncomingSentence*
  `public void IncomingSentence( EDU.rutgers.stimulate.MIM.RawNLIEvent` **sentence** `)`
  - Usage
    * called when NLI occurs. The input is presented as a Sentence object.
  - Parameters
    * `sentence` - the input in the form of an event.

2.1.6 *Interface* MIMPointer2DEventListener

Event listener interface for raw 2D pointer input events.

Declaration

public abstract interface MIMPointer2DEventListener

Methods

- *Pointer2DInput*
  `public void Pointer2DInput( EDU.rutgers.stimulate.MIM.RawPointer2DEvent` **event** `)`
  - Usage
    - called when NLI occurs. The input is presented as a Sentence object.
  - Parameters
    - `sentence` - the input in the form of an event.
- *drawsCursor*
  `public boolean drawsCursor( EDU.rutgers.stimulate.MIM.RawPointer2DEvent` **event** `)`
  - Usage
    - Indicates whether the application will draw the cursor by itself or not.
  - Parameters
    - `event` - the event containing the coordinates.
  - Returns - boolean return true when the application draws the cursor itself.

2.1.7 *Interface* MIPointer

Interface for a Modal Input pointer object.

Declaration

public abstract interface MIPointer
implements MI

Methods

- *setCursorVisible*
  `public void setCursorVisible( boolean` **show** `)`
  - Usage
    - Shows or hides the cursor.
  - Parameters
    - `show` - set to true if you wish to show the cursor.
- *setDraggedComponent*
  `public void setDraggedComponent( java.awt.Component` **victim** `)`
  - Usage
    - Sets the component that is dragged by this input.
  - Parameters
    - `victim` - the component that is to be dragged by this input. May be set to null to reset the dragged component.
- *getDraggedComponent*
  `public Component getDraggedComponent( )`
  - Usage
    - Gets the component that is dragged by this input.
  - Returns - Component the component that is currently dragged by this input. May be null if no component is dragged.
- *setDraggedContainer*
  `public void setDraggedContainer( java.awt.Container` **victim** `)`

  - Usage
    - Sets the container of the component that is dragged by this input.
  - Parameters
    - `victim` - the component that is to be dragged by this input. May be set to null to reset the dragged component.

- *getDraggedContainer*
  `public Container getDraggedContainer( )`
  - Usage
    - Gets the container of the component that is dragged by this input.
  - Returns - Container the container of the component that is currently dragged by this input. May be null if no component is dragged.

2.1.8 *Interface* NLFeedback

Interface for providing NL feedback.

Declaration

public abstract interface NLFeedback

Methods

- *outputFeedback*
  `public boolean outputFeedback( java.lang.String sentence )`
  - Usage
    - Outputs this string.
  - Parameters
    - `sentence` - the output in the form of a natural sentence.
  - Returns - boolean true if it went right, false if it went wrong.

2.1.9 *Interface* NLISentenceListener

Event listener interface for Natural Language Interface Sentence events.

Declaration

public abstract interface NLISentenceListener

Methods

- *getApp*
  `public MIApp getApp( )`
  - Usage
    - Gets the app associated with the listener.
  - Returns - MIApp the app object associated with this listener.

- *NLIAsSentence*
  `public String NLIAsSentence( EDU.rutgers.stimulate.MIM.NLISentenceEvent sentence )`
  - Usage

    * called when NLI occurs. The actual input is presented in Sentence format just as the sentence which was recognized. There is no information about confidence measures or per-word timestamps.
  - Parameters
    * `sentence` - the input in the form of an event.
  - Returns - string the feedback if the presented sentence was invalid. This MAY cause the manager to send the feedback to the user, but also to provide the app object another alternative or present the sentence to another app object.

2.1.10 *Interface* NLIStringListener

Event listener interface for Natural Language Interface String events.

Declaration

```
public abstract interface NLIStringListener
```

Methods

- *getApp*
  `public MIApp getApp( )`
  - Usage
    * Gets the app associated with the listener.
  - Returns - MIApp the app object associated with this listener.
- *NLIAsString*
  `public String NLIAsString( EDU.rutgers.stimulate.MIM.NLIStringEvent sentence )`
  - Usage
    * called when NLI occurs. The actual input is presented in String format just as the sentence which was recognized. There is no information about confidence measures or per-word timestamps.
  - Parameters
    * `sentence` - the input in the form of an event.
  - Returns - String the feedback if the presented sentence was invalid. This may cause the manager to send the feedback to the human app, but also to provide the app object another alternative or present the sentence to another app object.

2.1.11 *Interface* TimedEvent

Basic timestamps interface to add to events.

Declaration

```
public abstract interface TimedEvent
```

Methods

- *getWhen*
  `public long getWhen( )`
  - Usage
    * Retrieves the time when the event happened. Time is measured in milliseconds, like the System.currentTimeMillis() method.
  - Returns - time that the event occurred, in milliseconds since 1970.

2.1.12 *Interface* ForceFeedback

Interface for providing Force feedback.

Declaration

public abstract interface ForceFeedback

Methods

- *forceFeedback*
  `public boolean forceFeedback( float force )`
  - Usage
    - Outputs this string.
  - Parameters
    - `force` - the normalized force feedback. Ranges from 0.0f to 1.0f.
  - Returns - boolean true if it went right, false if it went wrong.

2.2 Classes

2.2.1 *Class* ConnectPanel

Declaration

public class ConnectPanel
extends javax.swing.JPanel

Serializable Fields

- private String typedTextName
  –
- private String typedTextPort
  –
- private JTextField textFieldName
  –
- private JTextField textFieldPort
  –
- private int port
  –

Constructors

- *ConnectPanel*
  `public ConnectPanel( java.lang.String namestr, java.lang.String portstr )`

Methods

- *getName*
  `public String getName( )`
- *getPort*
  `public int getPort( )`

Methods inherited from class `javax.swing.JPanel`

- `public AccessibleContext getAccessibleContext( )`
- `public String getUIClassID( )`
- `protected String paramString( )`
- `public void updateUI( )`

Methods inherited from class `javax.swing.JComponent`

- `public void addAncestorListener( javax.swing.event.AncestorListener )`
- `public void addNotify( )`
- `public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )`
- `public void computeVisibleRect( java.awt.Rectangle )`
- `public boolean contains( int , int )`
- `public JToolTip createToolTip( )`
- `public void firePropertyChange( java.lang.String , byte , byte )`
- `public void firePropertyChange( java.lang.String , char , char )`
- `public void firePropertyChange( java.lang.String , double , double )`
- `public void firePropertyChange( java.lang.String , float , float )`
- `public void firePropertyChange( java.lang.String , int , int )`
- `public void firePropertyChange( java.lang.String , long , long )`
- `protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )`
- `public void firePropertyChange( java.lang.String , short , short )`
- `public void firePropertyChange( java.lang.String , boolean , boolean )`
- `protected void fireVetoableChange( java.lang.String , java.lang.Object , java.lang.Object )`
- `public AccessibleContext getAccessibleContext( )`
- `public ActionListener getActionForKeyStroke( javax.swing.KeyStroke )`
- `public float getAlignmentX( )`
- `public float getAlignmentY( )`
- `public boolean getAutoscrolls( )`
- `public Border getBorder( )`
- `public Rectangle getBounds( java.awt.Rectangle )`
- `public final Object getClientProperty( java.lang.Object )`
- `protected Graphics getComponentGraphics( java.awt.Graphics )`
- `public int getConditionForKeyStroke( javax.swing.KeyStroke )`
- `public int getDebugGraphicsOptions( )`
- `public Graphics getGraphics( )`
- `public int getHeight( )`
- `public Insets getInsets( )`
- `public Insets getInsets( java.awt.Insets )`
- `public Point getLocation( java.awt.Point )`
- `public Dimension getMaximumSize( )`
- `public Dimension getMinimumSize( )`
- `public Component getNextFocusableComponent( )`
- `public Dimension getPreferredSize( )`
- `public KeyStroke getRegisteredKeyStrokes( )`
- `public JRootPane getRootPane( )`
- `public Dimension getSize( java.awt.Dimension )`

- `public Point getToolTipLocation( java.awt.event.MouseEvent )`
- `public String getToolTipText( )`
- `public String getToolTipText( java.awt.event.MouseEvent )`
- `public Container getTopLevelAncestor( )`
- `public String getUIClassID( )`
- `public Rectangle getVisibleRect( )`
- `public int getWidth( )`
- `public int getX( )`
- `public int getY( )`
- `public void grabFocus( )`
- `public boolean hasFocus( )`
- `public boolean isDoubleBuffered( )`
- `public boolean isFocusCycleRoot( )`
- `public boolean isFocusTraversable( )`
- `public static boolean isLightweightComponent( java.awt.Component )`
- `public boolean isManagingFocus( )`
- `public boolean isOpaque( )`
- `public boolean isOptimizedDrawingEnabled( )`
- `public boolean isPaintingTile( )`
- `public boolean isRequestFocusEnabled( )`
- `public boolean isValidateRoot( )`
- `public void paint( java.awt.Graphics )`
- `protected void paintBorder( java.awt.Graphics )`
- `protected void paintChildren( java.awt.Graphics )`
- `protected void paintComponent( java.awt.Graphics )`
- `public void paintImmediately( int , int , int , int )`
- `public void paintImmediately( java.awt.Rectangle )`
- `protected String paramString( )`
- `protected void processComponentKeyEvent( java.awt.event.KeyEvent )`
- `protected void processFocusEvent( java.awt.event.FocusEvent )`
- `protected void processKeyEvent( java.awt.event.KeyEvent )`
- `protected void processMouseMotionEvent( java.awt.event.MouseEvent )`
- `public final void putClientProperty( java.lang.Object , java.lang.Object )`
- `public void registerKeyboardAction( java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int )`
- `public void registerKeyboardAction( java.awt.event.ActionListener , javax.swing.KeyStroke , int )`
- `public void removeAncestorListener( javax.swing.event.AncestorListener )`
- `public void removeNotify( )`
- `public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void removeVetoableChangeListener( java.beans.VetoableChangeListener )`
- `public void repaint( long , int , int , int , int )`
- `public void repaint( java.awt.Rectangle )`
- `public boolean requestDefaultFocus( )`
- `public void requestFocus( )`
- `public void resetKeyboardActions( )`
- `public void reshape( int , int , int , int )`
- `public void revalidate( )`
- `public void scrollRectToVisible( java.awt.Rectangle )`
- `public void setAlignmentX( float )`
- `public void setAlignmentY( float )`
- `public void setAutoscrolls( boolean )`
- `public void setBackground( java.awt.Color )`
- `public void setBorder( javax.swing.border.Border )`
- `public void setDebugGraphicsOptions( int )`
- `public void setDoubleBuffered( boolean )`
- `public void setEnabled( boolean )`
- `public void setFont( java.awt.Font )`
- `public void setForeground( java.awt.Color )`
- `public void setMaximumSize( java.awt.Dimension )`

- public void setMinimumSize(java.awt.Dimension)
- public void setNextFocusableComponent(java.awt.Component)
- public void setOpaque(boolean)
- public void setPreferredSize(java.awt.Dimension)
- public void setRequestFocusEnabled(boolean)
- public void setToolTipText(java.lang.String)
- protected void setUI(javax.swing.plaf.ComponentUI)
- public void setVisible(boolean)
- public void unregisterKeyboardAction(javax.swing.KeyStroke)
- public void update(java.awt.Graphics)
- public void updateUI( )

Methods inherited from class `java.awt.Container`

- public Component add(java.awt.Component)
- public Component add(java.awt.Component , int)
- public void add(java.awt.Component , java.lang.Object)
- public void add(java.awt.Component , java.lang.Object , int)
- public Component add(java.lang.String , java.awt.Component)
- public synchronized void addContainerListener(java.awt.event.ContainerListener)
- protected void addImpl(java.awt.Component , java.lang.Object , int)
- public void addNotify( )
- public int countComponents( )
- public void deliverEvent(java.awt.Event)
- public void doLayout( )
- public Component findComponentAt(int , int)
- public Component findComponentAt(java.awt.Point)
- public float getAlignmentX( )
- public float getAlignmentY( )
- public Component getComponent(int)
- public Component getComponentAt(int , int)
- public Component getComponentAt(java.awt.Point)
- public int getComponentCount( )
- public Component getComponents( )
- public Insets getInsets( )
- public LayoutManager getLayout( )
- public Dimension getMaximumSize( )
- public Dimension getMinimumSize( )
- public Dimension getPreferredSize( )
- public Insets insets( )
- public void invalidate( )
- public boolean isAncestorOf(java.awt.Component)
- public void layout( )
- public void list(java.io.PrintStream , int)
- public void list(java.io.PrintWriter , int)
- public Component locate(int , int)
- public Dimension minimumSize( )
- public void paint(java.awt.Graphics)
- public void paintComponents(java.awt.Graphics)
- protected String paramString( )
- public Dimension preferredSize( )
- public void print(java.awt.Graphics)
- public void printComponents(java.awt.Graphics)
- protected void processContainerEvent(java.awt.event.ContainerEvent)
- protected void processEvent(java.awt.AWTEvent)
- public void remove(int)
- public void remove(java.awt.Component)
- public void removeAll( )
- public synchronized void removeContainerListener(java.awt.event.ContainerListener)
- public void removeNotify( )
- public void setFont(java.awt.Font)
- public void setLayout(java.awt.LayoutManager)
- public void update(java.awt.Graphics)
- public void validate( )
- protected void validateTree( )

Methods inherited from class `java.awt.Component`

- `public boolean action( java.awt.Event , java.lang.Object )`
- `public synchronized void add( java.awt.PopupMenu )`
- `public synchronized void addComponentListener( java.awt.event.ComponentListener )`
- `public synchronized void addFocusListener( java.awt.event.FocusListener )`
- `public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
- `public synchronized void addKeyListener( java.awt.event.KeyListener )`
- `public synchronized void addMouseListener( java.awt.event.MouseListener )`
- `public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
- `public void addNotify( )`
- `public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- `public Rectangle bounds( )`
- `public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )`
- `public int checkImage( java.awt.Image , java.awt.image.ImageObserver )`
- `protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )`
- `public boolean contains( int , int )`
- `public boolean contains( java.awt.Point )`
- `public Image createImage( int , int )`
- `public Image createImage( java.awt.image.ImageProducer )`
- `public void deliverEvent( java.awt.Event )`
- `public void disable( )`
- `protected final void disableEvents( long )`
- `public final void dispatchEvent( java.awt.AWTEvent )`
- `public void doLayout( )`
- `public void enable( )`
- `public void enable( boolean )`
- `protected final void enableEvents( long )`
- `public void enableInputMethods( boolean )`
- `protected void firePropertyChange( java.lang.String , java.lang.Object , java.lang.Object )`
- `public float getAlignmentX( )`
- `public float getAlignmentY( )`
- `public Color getBackground( )`
- `public Rectangle getBounds( )`
- `public Rectangle getBounds( java.awt.Rectangle )`
- `public ColorModel getColorModel( )`
- `public Component getComponentAt( int , int )`
- `public Component getComponentAt( java.awt.Point )`
- `public ComponentOrientation getComponentOrientation( )`
- `public Cursor getCursor( )`
- `public synchronized DropTarget getDropTarget( )`
- `public Font getFont( )`
- `public FontMetrics getFontMetrics( java.awt.Font )`
- `public Color getForeground( )`
- `public Graphics getGraphics( )`
- `public int getHeight( )`
- `public InputContext getInputContext( )`
- `public InputMethodRequests getInputMethodRequests( )`
- `public Locale getLocale( )`
- `public Point getLocation( )`
- `public Point getLocation( java.awt.Point )`
- `public Point getLocationOnScreen( )`
- `public Dimension getMaximumSize( )`
- `public Dimension getMinimumSize( )`
- `public String getName( )`
- `public Container getParent( )`
- `public ComponentPeer getPeer( )`

- public Dimension getPreferredSize( )
- public Dimension getSize( )
- public Dimension getSize(java.awt.Dimension)
- public Toolkit getToolkit( )
- public final Object getTreeLock( )
- public int getWidth( )
- public int getX( )
- public int getY( )
- public boolean gotFocus(java.awt.Event , java.lang.Object)
- public boolean handleEvent(java.awt.Event)
- public boolean hasFocus( )
- public void hide( )
- public boolean imageUpdate(java.awt.Image , int , int , int , int , int)
- public boolean inside(int , int)
- public void invalidate( )
- public boolean isDisplayable( )
- public boolean isDoubleBuffered( )
- public boolean isEnabled( )
- public boolean isFocusTraversable( )
- public boolean isLightweight( )
- public boolean isOpaque( )
- public boolean isShowing( )
- public boolean isValid( )
- public boolean isVisible( )
- public boolean keyDown(java.awt.Event , int)
- public boolean keyUp(java.awt.Event , int)
- public void layout( )
- public void list( )
- public void list(java.io.PrintStream)
- public void list(java.io.PrintStream , int)
- public void list(java.io.PrintWriter)
- public void list(java.io.PrintWriter , int)
- public Component locate(int , int)
- public Point location( )
- public boolean lostFocus(java.awt.Event , java.lang.Object)
- public Dimension minimumSize( )
- public boolean mouseDown(java.awt.Event , int , int)
- public boolean mouseDrag(java.awt.Event , int , int)
- public boolean mouseEnter(java.awt.Event , int , int)
- public boolean mouseExit(java.awt.Event , int , int)
- public boolean mouseMove(java.awt.Event , int , int)
- public boolean mouseUp(java.awt.Event , int , int)
- public void move(int , int)
- public void nextFocus( )
- public void paint(java.awt.Graphics)
- public void paintAll(java.awt.Graphics)
- protected String paramString( )
- public boolean postEvent(java.awt.Event)
- public Dimension preferredSize( )
- public boolean prepareImage(java.awt.Image , int , int , java.awt.image.ImageObserver)
- public boolean prepareImage(java.awt.Image , java.awt.image.ImageObserver)
- public void print(java.awt.Graphics)
- public void printAll(java.awt.Graphics)
- protected void processComponentEvent(java.awt.event.ComponentEvent)
- protected void processEvent(java.awt.AWTEvent)
- protected void processFocusEvent(java.awt.event.FocusEvent)
- protected void processInputMethodEvent(java.awt.event.InputMethodEvent)
- protected void processKeyEvent(java.awt.event.KeyEvent)
- protected void processMouseEvent(java.awt.event.MouseEvent)
- protected void processMouseMotionEvent(java.awt.event.MouseEvent)

- `public synchronized void remove( java.awt.MenuComponent )`
- `public synchronized void removeComponentListener( java.awt.event.ComponentListener )`
- `public synchronized void removeFocusListener( java.awt.event.FocusListener )`
- `public synchronized void removeInputMethodListener( java.awt.event.InputMethodListener )`
- `public synchronized void removeKeyListener( java.awt.event.KeyListener )`
- `public synchronized void removeMouseListener( java.awt.event.MouseListener )`
- `public synchronized void removeMouseMotionListener( java.awt.event.MouseMotionListener )`
- `public void removeNotify( )`
- `public synchronized void removePropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void removePropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- `public void repaint( )`
- `public void repaint( int , int , int , int )`
- `public void repaint( long )`
- `public void repaint( long , int , int , int , int )`
- `public void requestFocus( )`
- `public void reshape( int , int , int , int )`
- `public void resize( int , int )`
- `public void resize( java.awt.Dimension )`
- `public void setBackground( java.awt.Color )`
- `public void setBounds( int , int , int , int )`
- `public void setBounds( java.awt.Rectangle )`
- `public void setComponentOrientation( java.awt.ComponentOrientation )`
- `public synchronized void setCursor( java.awt.Cursor )`
- `public synchronized void setDropTarget( java.awt.dnd.DropTarget )`
- `public void setEnabled( boolean )`
- `public void setFont( java.awt.Font )`
- `public void setForeground( java.awt.Color )`
- `public void setLocale( java.util.Locale )`
- `public void setLocation( int , int )`
- `public void setLocation( java.awt.Point )`
- `public void setName( java.lang.String )`
- `public void setSize( int , int )`
- `public void setSize( java.awt.Dimension )`
- `public void setVisible( boolean )`
- `public void show( )`
- `public void show( boolean )`
- `public Dimension size( )`
- `public String toString( )`
- `public void transferFocus( )`
- `public void update( java.awt.Graphics )`
- `public void validate( )`

2.2.2 *Class* Directional3DEvent

Event class for 3D pointer events with both position and orientation.

Declaration

public class Directional3DEvent
extends EDU.rutgers.stimulate.MIM.SimplePointer3DEvent

Serializable Fields

- private final Vector3d direction3d
  –

Constructors

- *Directional3DEvent*
  `public Directional3DEvent( java.awt.Component` **source**`, long` **when**`, double` **x**`, double` **y**`, javax.vecmath.Point3d` **point**`, javax.vecmath.Vector3d` **direction** `)`
  - Usage
    - Normal constructor, with the coordinates.
  - Parameters
    - `source` - the source object.
    - `when` - the timestamp.
    - `x,y` - the point where the event occurred.

Methods

- *getDirection3D*
  `public Vector3d getDirection3D( )`
  - Usage
    - Get the orientation of this event.
  - Returns - Vector3d the 3D vector describing the orientation of the event.

Methods inherited from class `EDU.rutgers.stimulate.MIM.SimplePointer3DEvent`

(in 2.2.13, page 44)

- `public Point3d getPoint3D( )`
- `public Point3d getOriginalPoint3D( )`
- `public void adjustPoint( javax.vecmath.Point3d` **newPoint** `)`

Methods inherited from class `EDU.rutgers.stimulate.MIM.RawPointer2DEvent`

(in 2.2.12, page 43)

- `public Point getPoint( )`
- `public double getX( )`
- `public double getY( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.3 *Class* Grammar

Specifies the grammar for an application. This class can read files in the Microsoft Speech SDK 4.0 grammar source format, and enables the manager to extend the grammar with it's own grammar (a unification of the formal languages conforming to the grammars).

Declaration

public class Grammar
extends java.lang.Object

Constructors

- *Grammar*
  `public Grammar( )`
  - Usage
    * Empty contructor.
- *Grammar*
  `public Grammar( java.net.URL` **`source`** `)`
  - Usage
    * Constructs a new Grammar object from a MS SSDK 4.0 grammar source file given as a URL object.
  - Parameters
    * `source` - the URL source object.
- *Grammar*
  `public Grammar( java.io.BufferedReader` **`source`** `)`
  - Usage
    * Constructs a new Grammar object from a MS SSDK 4.0 grammar source file given as a BufferedReader object.
  - Parameters
    * `source` - the BufferedReader source object.

Methods

- *getGrammarData*
  `public byte getGrammarData( )`
  - Usage
    * Get the grammar source.
  - Returns - byte[] an array of bytes with the sourcecode of the grammar in MS Speech SDK 4.0 text format.
- *setAdditionalRules*
  `public void setAdditionalRules( java.lang.String` **`sectionName`**`, java.lang.String` **`rules`** `)`

2.2.4 *Class* HandMotionEvent

Event class for hand motion.

Declaration

public class HandMotionEvent
extends EDU.rutgers.stimulate.MIM.Directional3DEvent

Serializable Fields

- private final GVector flexVector
  –

Constructors

- *HandMotionEvent*
  `public HandMotionEvent( java.awt.Component` **source**, `long` **when**, `double` **x**, `double` **y**, `javax.vecmath.Point3d` **point**, `javax.vecmath.Vector3d` **direction**, `javax.vecmath.GVector` **flexing** `)`
  - Usage
    * Normal constructor, with the coordinates.
  - Parameters
    * `source` - the source object.
    * `when` - the timestamp.
    * `x,y` - the point where the event occurred.

Methods

- *getFingerFlexing*
  `public GVector getFingerFlexing( )`
  - Usage
    * Get the finger flexing.
  - Returns - GVector the n-D vector describing the flexing of the fingers.

Methods inherited from class `EDU.rutgers.stimulate.MIM.Directional3DEvent`

(in 2.2.2, page 27)

- `public Vector3d getDirection3D( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.SimplePointer3DEvent`

(in 2.2.13, page 44)

- `public Point3d getPoint3D( )`
- `public Point3d getOriginalPoint3D( )`
- `public void adjustPoint( javax.vecmath.Point3d` **newPoint** `)`

Methods inherited from class `EDU.rutgers.stimulate.MIM.RawPointer2DEvent`

(in 2.2.12, page 43)

- `public Point getPoint( )`
- `public double getX( )`
- `public double getY( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.5 *Class* JCursor

Generic on-screen non-changable cursor.

Declaration

public class JCursor
extends javax.swing.JComponent

Constructors

- *JCursor*
  `public JCursor( java.lang.String` **imagePath** `)`
- *JCursor*
  `public JCursor( java.net.URL` **imageURL** `)`

Methods

- *paint*
  `public void paint( java.awt.Graphics` **g** `)`
- *contains*
  `public boolean contains( int` **x**, `int` **y** `)`
- *contains*
  `public boolean contains( java.awt.Point` **p** `)`
- *setLocation*
  `public void setLocation( int` **x**, `int` **y** `)`

Methods inherited from class `javax.swing.JComponent`

- `public void addAncestorListener( javax.swing.event.AncestorListener )`
- `public void addNotify( )`
- `public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void addVetoableChangeListener( java.beans.VetoableChangeListener )`
- `public void computeVisibleRect( java.awt.Rectangle )`
- `public boolean contains( int , int )`
- `public JToolTip createToolTip( )`
- `public void firePropertyChange( java.lang.String , byte , byte )`
- `public void firePropertyChange( java.lang.String , char , char )`
- `public void firePropertyChange( java.lang.String , double , double )`
- `public void firePropertyChange( java.lang.String , float , float )`

- public void firePropertyChange(java.lang.String , int , int)
- public void firePropertyChange(java.lang.String , long , long)
- protected void firePropertyChange(java.lang.String , java.lang.Object , java.lang.Object)
- public void firePropertyChange(java.lang.String , short , short)
- public void firePropertyChange(java.lang.String , boolean , boolean)
- protected void fireVetoableChange(java.lang.String , java.lang.Object , java.lang.Object)
- public AccessibleContext getAccessibleContext( )
- public ActionListener getActionForKeyStroke(javax.swing.KeyStroke)
- public float getAlignmentX( )
- public float getAlignmentY( )
- public boolean getAutoscrolls( )
- public Border getBorder( )
- public Rectangle getBounds(java.awt.Rectangle)
- public final Object getClientProperty(java.lang.Object)
- protected Graphics getComponentGraphics(java.awt.Graphics)
- public int getConditionForKeyStroke(javax.swing.KeyStroke)
- public int getDebugGraphicsOptions( )
- public Graphics getGraphics( )
- public int getHeight( )
- public Insets getInsets( )
- public Insets getInsets(java.awt.Insets)
- public Point getLocation(java.awt.Point)
- public Dimension getMaximumSize( )
- public Dimension getMinimumSize( )
- public Component getNextFocusableComponent( )
- public Dimension getPreferredSize( )
- public KeyStroke getRegisteredKeyStrokes( )
- public JRootPane getRootPane( )
- public Dimension getSize(java.awt.Dimension)
- public Point getToolTipLocation(java.awt.event.MouseEvent)
- public String getToolTipText( )
- public String getToolTipText(java.awt.event.MouseEvent)
- public Container getTopLevelAncestor( )
- public String getUIClassID( )
- public Rectangle getVisibleRect( )
- public int getWidth( )
- public int getX( )
- public int getY( )
- public void grabFocus( )
- public boolean hasFocus( )
- public boolean isDoubleBuffered( )
- public boolean isFocusCycleRoot( )
- public boolean isFocusTraversable( )
- public static boolean isLightweightComponent(java.awt.Component)
- public boolean isManagingFocus( )
- public boolean isOpaque( )
- public boolean isOptimizedDrawingEnabled( )
- public boolean isPaintingTile( )
- public boolean isRequestFocusEnabled( )
- public boolean isValidateRoot( )
- public void paint(java.awt.Graphics)
- protected void paintBorder(java.awt.Graphics)
- protected void paintChildren(java.awt.Graphics)
- protected void paintComponent(java.awt.Graphics)
- public void paintImmediately(int , int , int , int)
- public void paintImmediately(java.awt.Rectangle)
- protected String paramString( )
- protected void processComponentKeyEvent(java.awt.event.KeyEvent)
- protected void processFocusEvent(java.awt.event.FocusEvent)
- protected void processKeyEvent(java.awt.event.KeyEvent)

- protected void processMouseMotionEvent(java.awt.event.MouseEvent)
- public final void putClientProperty(java.lang.Object , java.lang.Object)
- public void registerKeyboardAction(java.awt.event.ActionListener , java.lang.String , javax.swing.KeyStroke , int)
- public void registerKeyboardAction(java.awt.event.ActionListener , javax.swing.KeyStroke , int)
- public void removeAncestorListener(javax.swing.event.AncestorListener)
- public void removeNotify( )
- public synchronized void removePropertyChangeListener(java.beans.PropertyChangeListener)
- public synchronized void removeVetoableChangeListener(java.beans.VetoableChangeListener)
- public void repaint(long , int , int , int , int)
- public void repaint(java.awt.Rectangle)
- public boolean requestDefaultFocus( )
- public void requestFocus( )
- public void resetKeyboardActions( )
- public void reshape(int , int , int , int)
- public void revalidate( )
- public void scrollRectToVisible(java.awt.Rectangle)
- public void setAlignmentX(float)
- public void setAlignmentY(float)
- public void setAutoscrolls(boolean)
- public void setBackground(java.awt.Color)
- public void setBorder(javax.swing.border.Border)
- public void setDebugGraphicsOptions(int)
- public void setDoubleBuffered(boolean)
- public void setEnabled(boolean)
- public void setFont(java.awt.Font)
- public void setForeground(java.awt.Color)
- public void setMaximumSize(java.awt.Dimension)
- public void setMinimumSize(java.awt.Dimension)
- public void setNextFocusableComponent(java.awt.Component)
- public void setOpaque(boolean)
- public void setPreferredSize(java.awt.Dimension)
- public void setRequestFocusEnabled(boolean)
- public void setToolTipText(java.lang.String)
- protected void setUI(javax.swing.plaf.ComponentUI)
- public void setVisible(boolean)
- public void unregisterKeyboardAction(javax.swing.KeyStroke)
- public void update(java.awt.Graphics)
- public void updateUI( )

Methods inherited from class `java.awt.Container`

- public Component add(java.awt.Component)
- public Component add(java.awt.Component , int)
- public void add(java.awt.Component , java.lang.Object)
- public void add(java.awt.Component , java.lang.Object , int)
- public Component add(java.lang.String , java.awt.Component)
- public synchronized void addContainerListener(java.awt.event.ContainerListener)
- protected void addImpl(java.awt.Component , java.lang.Object , int)
- public void addNotify( )
- public int countComponents( )
- public void deliverEvent(java.awt.Event)
- public void doLayout( )
- public Component findComponentAt(int , int)
- public Component findComponentAt(java.awt.Point)
- public float getAlignmentX( )
- public float getAlignmentY( )
- public Component getComponent(int)

- `public Component getComponentAt( int , int )`
- `public Component getComponentAt( java.awt.Point )`
- `public int getComponentCount( )`
- `public Component getComponents( )`
- `public Insets getInsets( )`
- `public LayoutManager getLayout( )`
- `public Dimension getMaximumSize( )`
- `public Dimension getMinimumSize( )`
- `public Dimension getPreferredSize( )`
- `public Insets insets( )`
- `public void invalidate( )`
- `public boolean isAncestorOf( java.awt.Component )`
- `public void layout( )`
- `public void list( java.io.PrintStream , int )`
- `public void list( java.io.PrintWriter , int )`
- `public Component locate( int , int )`
- `public Dimension minimumSize( )`
- `public void paint( java.awt.Graphics )`
- `public void paintComponents( java.awt.Graphics )`
- `protected String paramString( )`
- `public Dimension preferredSize( )`
- `public void print( java.awt.Graphics )`
- `public void printComponents( java.awt.Graphics )`
- `protected void processContainerEvent( java.awt.event.ContainerEvent )`
- `protected void processEvent( java.awt.AWTEvent )`
- `public void remove( int )`
- `public void remove( java.awt.Component )`
- `public void removeAll( )`
- `public synchronized void removeContainerListener( java.awt.event.ContainerListener )`
- `public void removeNotify( )`
- `public void setFont( java.awt.Font )`
- `public void setLayout( java.awt.LayoutManager )`
- `public void update( java.awt.Graphics )`
- `public void validate( )`
- `protected void validateTree( )`

Methods inherited from class `java.awt.Component`

- `public boolean action( java.awt.Event , java.lang.Object )`
- `public synchronized void add( java.awt.PopupMenu )`
- `public synchronized void addComponentListener( java.awt.event.ComponentListener )`
- `public synchronized void addFocusListener( java.awt.event.FocusListener )`
- `public synchronized void addInputMethodListener( java.awt.event.InputMethodListener )`
- `public synchronized void addKeyListener( java.awt.event.KeyListener )`
- `public synchronized void addMouseListener( java.awt.event.MouseListener )`
- `public synchronized void addMouseMotionListener( java.awt.event.MouseMotionListener )`
- `public void addNotify( )`
- `public synchronized void addPropertyChangeListener( java.beans.PropertyChangeListener )`
- `public synchronized void addPropertyChangeListener( java.lang.String , java.beans.PropertyChangeListener )`
- `public Rectangle bounds( )`
- `public int checkImage( java.awt.Image , int , int , java.awt.image.ImageObserver )`
- `public int checkImage( java.awt.Image , java.awt.image.ImageObserver )`
- `protected AWTEvent coalesceEvents( java.awt.AWTEvent , java.awt.AWTEvent )`
- `public boolean contains( int , int )`
- `public boolean contains( java.awt.Point )`
- `public Image createImage( int , int )`
- `public Image createImage( java.awt.image.ImageProducer )`
- `public void deliverEvent( java.awt.Event )`

- public void disable( )
- protected final void disableEvents(long)
- public final void dispatchEvent(java.awt.AWTEvent)
- public void doLayout( )
- public void enable( )
- public void enable(boolean)
- protected final void enableEvents(long)
- public void enableInputMethods(boolean)
- protected void firePropertyChange(java.lang.String , java.lang.Object , java.lang.Object)
- public float getAlignmentX( )
- public float getAlignmentY( )
- public Color getBackground( )
- public Rectangle getBounds( )
- public Rectangle getBounds(java.awt.Rectangle)
- public ColorModel getColorModel( )
- public Component getComponentAt(int , int)
- public Component getComponentAt(java.awt.Point)
- public ComponentOrientation getComponentOrientation( )
- public Cursor getCursor( )
- public synchronized DropTarget getDropTarget( )
- public Font getFont( )
- public FontMetrics getFontMetrics(java.awt.Font)
- public Color getForeground( )
- public Graphics getGraphics( )
- public int getHeight( )
- public InputContext getInputContext( )
- public InputMethodRequests getInputMethodRequests( )
- public Locale getLocale( )
- public Point getLocation( )
- public Point getLocation(java.awt.Point)
- public Point getLocationOnScreen( )
- public Dimension getMaximumSize( )
- public Dimension getMinimumSize( )
- public String getName( )
- public Container getParent( )
- public ComponentPeer getPeer( )
- public Dimension getPreferredSize( )
- public Dimension getSize( )
- public Dimension getSize(java.awt.Dimension)
- public Toolkit getToolkit( )
- public final Object getTreeLock( )
- public int getWidth( )
- public int getX( )
- public int getY( )
- public boolean gotFocus(java.awt.Event , java.lang.Object)
- public boolean handleEvent(java.awt.Event)
- public boolean hasFocus( )
- public void hide( )
- public boolean imageUpdate(java.awt.Image , int , int , int , int , int)
- public boolean inside(int , int)
- public void invalidate( )
- public boolean isDisplayable( )
- public boolean isDoubleBuffered( )
- public boolean isEnabled( )
- public boolean isFocusTraversable( )
- public boolean isLightweight( )
- public boolean isOpaque( )
- public boolean isShowing( )
- public boolean isValid( )
- public boolean isVisible( )

- public boolean keyDown(java.awt.Event , int)
- public boolean keyUp(java.awt.Event , int)
- public void layout( )
- public void list( )
- public void list(java.io.PrintStream)
- public void list(java.io.PrintStream , int)
- public void list(java.io.PrintWriter)
- public void list(java.io.PrintWriter , int)
- public Component locate(int , int)
- public Point location( )
- public boolean lostFocus(java.awt.Event , java.lang.Object)
- public Dimension minimumSize( )
- public boolean mouseDown(java.awt.Event , int , int)
- public boolean mouseDrag(java.awt.Event , int , int)
- public boolean mouseEnter(java.awt.Event , int , int)
- public boolean mouseExit(java.awt.Event , int , int)
- public boolean mouseMove(java.awt.Event , int , int)
- public boolean mouseUp(java.awt.Event , int , int)
- public void move(int , int)
- public void nextFocus( )
- public void paint(java.awt.Graphics)
- public void paintAll(java.awt.Graphics)
- protected String paramString( )
- public boolean postEvent(java.awt.Event)
- public Dimension preferredSize( )
- public boolean prepareImage(java.awt.Image , int , int , java.awt.image.ImageObserver)
- public boolean prepareImage(java.awt.Image , java.awt.image.ImageObserver)
- public void print(java.awt.Graphics)
- public void printAll(java.awt.Graphics)
- protected void processComponentEvent(java.awt.event.ComponentEvent)
- protected void processEvent(java.awt.AWTEvent)
- protected void processFocusEvent(java.awt.event.FocusEvent)
- protected void processInputMethodEvent(java.awt.event.InputMethodEvent)
- protected void processKeyEvent(java.awt.event.KeyEvent)
- protected void processMouseEvent(java.awt.event.MouseEvent)
- protected void processMouseMotionEvent(java.awt.event.MouseEvent)
- public synchronized void remove(java.awt.MenuComponent)
- public synchronized void removeComponentListener(java.awt.event.ComponentListener)
- public synchronized void removeFocusListener(java.awt.event.FocusListener)
- public synchronized void removeInputMethodListener(java.awt.event.InputMethodListener)
- public synchronized void removeKeyListener(java.awt.event.KeyListener)
- public synchronized void removeMouseListener(java.awt.event.MouseListener)
- public synchronized void removeMouseMotionListener(java.awt.event.MouseMotionListener)
- public void removeNotify( )
- public synchronized void removePropertyChangeListener(java.beans.PropertyChangeListener)
- public synchronized void removePropertyChangeListener(java.lang.String , java.beans.PropertyChangeListener)
- public void repaint( )
- public void repaint(int , int , int , int)
- public void repaint(long)
- public void repaint(long , int , int , int , int)
- public void requestFocus( )
- public void reshape(int , int , int , int)
- public void resize(int , int)
- public void resize(java.awt.Dimension)
- public void setBackground(java.awt.Color)
- public void setBounds(int , int , int , int)
- public void setBounds(java.awt.Rectangle)
- public void setComponentOrientation(java.awt.ComponentOrientation)
- public synchronized void setCursor(java.awt.Cursor)

- `public synchronized void setDropTarget( java.awt.dnd.DropTarget )`
- `public void setEnabled( boolean )`
- `public void setFont( java.awt.Font )`
- `public void setForeground( java.awt.Color )`
- `public void setLocale( java.util.Locale )`
- `public void setLocation( int , int )`
- `public void setLocation( java.awt.Point )`
- `public void setName( java.lang.String )`
- `public void setSize( int , int )`
- `public void setSize( java.awt.Dimension )`
- `public void setVisible( boolean )`
- `public void show( )`
- `public void show( boolean )`
- `public Dimension size( )`
- `public String toString( )`
- `public void transferFocus( )`
- `public void update( java.awt.Graphics )`
- `public void validate( )`

2.2.6 *Class* MIEvent

Basic event object for modal input events. It adds the notion of timestamps to events.

Declaration

public class MIEvent
extends java.util.EventObject
implements TimedEvent

Serializable Fields

- private long timestamp
  –

Constructors

- *MIEvent*
  `public MIEvent( java.awt.Component` **`source`**`, long` **`when`** `)`
  - Usage
    * Constructs a new Modal Input Event.
  - Parameters
    * `source` - the source Component.
    * `when` - the timestamp of the event.

Methods

- *getWhen*
  `public long getWhen( )`
  - Usage
    * Retrieves the time when the event happened. Time is measured in milliseconds, like the System.currentTimeMillis() method.
  - Returns - time that the event occurred, in milliseconds since 1970.

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.7 *Class* MIEventHistory

Helper class to keep a history of MIEvents.

Declaration

public class MIEventHistory
extends java.lang.Object

Constructors

- *MIEventHistory*
  `public MIEventHistory( long` **timeout** `)`
  - Usage
    * Constructs a new history object with the specified timeout.

Methods

- *addEvent*
  `public synchronized void addEvent( EDU.rutgers.stimulate.MIM.MIEvent` **event** `)`
- *getEventAt*
  `public synchronized MIEvent getEventAt( long` **when** `)`
- *pack*
  `public synchronized void pack( )`
- *removeEvent*
  `public synchronized boolean removeEvent( EDU.rutgers.stimulate.MIM.MIEvent` **event** `)`

2.2.8 *Class* MIMAdvertiser

The MIMAdvertiser helps the modal input manager and applications find each other.

It simply keeps track of all the MI managers available (probably none or just one) and MI-aware beans, asking managers to cintact the apps when needed (ie. at an subscribe or a registration).

While it is possible to create an instance of this class, it is no use other than to have a shortcut for it, since it entirely consists of static variables and methods.

Usually, MI-aware beans should just implement the MIApp interface and subscribe. When or once a MI manager is available, it will be notified through this interface.

Note: while apps are consistently called 'beans', the actual objects that are subscribed can be any object, serving as an intermediary.

Declaration

public class MIMAdvertiser
extends java.lang.Object

Fields

- public static final String version
  –

Constructors

- *MIMAdvertiser*
  `public MIMAdvertiser( )`

Methods

- *getManagers*
  `public static Enumeration getManagers( )`
  – Usage
    * Finds all available (registered) managers. While this can be used to find a manager and register events with it, MI-aware beans should typically just subscribe.
  – Returns - Enumeration of all managers.
- *registerManager*
  `public static boolean registerManager( EDU.rutgers.stimulate.MIM.MIManager` **`manager`** `)`
  – Usage
    * Registers a manager. Adds it to the list of registered managers, then notifies the apps through the MIApp interface that it started.
  – Parameters
    * `manager` - the manager to register.
  – Returns - true if registration succeeded, false if manager was already registered.
- *unregisterManager*
  `public static boolean unregisterManager( EDU.rutgers.stimulate.MIM.MIManager` **`manager`** `)`
  – Usage
    * Unregisters a manager. Removes it from the list of managers, then notifies all apps that it stopped.
  – Parameters
    * `manager` - the manager to unregister.
  – Returns - true when unregistration succeeded, false if the manager was not registered.

- *getApps*
  `public static Enumeration getApps( )`
  – Usage
    * Finds all (subscribed) MI-aware beans. It is usually not necessary to use this function, since new managers are notified of existing apps when they register.
  – Returns - Enumeration of all subscribed apps.
- *subscribe*
  `public static boolean subscribe( EDU.rutgers.stimulate.MIM.MIApp` **app** `)`
  – Usage
    * Subscribe a MI-aware app bean to manager advertisement services. Adds it to the list, then notifies it that all existing managers ”started”.
  – Returns - true if subscription succeeded, false if app was already subscribed.
- *unsubscribe*
  `public static boolean unsubscribe( EDU.rutgers.stimulate.MIM.MIApp` **app** `)`
  – Usage
    * Unsubscribe a MI-aware bean from manager advertisement services. Removes it from the list, then notifies it that all existing managers ”stopped”.
  – Returns - true if unsubscription succeeded, false if the app was not known.
- *spamApps*
  `protected static void spamApps( EDU.rutgers.stimulate.MIM.MIManager` **manager**`, boolean` **register** `)`
  – Usage
    * Notifies the apps that there is a manager added or removed. Not for normal use.
  – Parameters
    * `manager` - the manager that cause the change.
    * `register` - set to true if manager registered, false if unregistered.
- *spamApp*
  `protected static void spamApp( EDU.rutgers.stimulate.MIM.MIApp` **app**`, boolean` **subscribe** `)`
  – Usage
    * Notifies the [un]subscribing app of managers. Not for normal use.
  – Parameters
    * `app` - the app that just [un]subscribed.
    * `subscribe` - set to true if the app subscribed, false if the app unsubscribed.

2.2.9 *Class* NLISentenceEvent

Natural Language Input event, presented in Sentence format.

Declaration

public class NLISentenceEvent **extends** EDU.rutgers.stimulate.MIM.MIEvent

Serializable Fields

- private Sentence sentence
  –
- private long duration
  –

Constructors

- *NLISentenceEvent*
  `public NLISentenceEvent( java.awt.Component` **source**`, long` **when**`, long` **duration**`,`
  `EDU.rutgers.stimulate.MIM.SpeechParsing.Sentence` **sentence** `)`
- *NLISentenceEvent*
  `public NLISentenceEvent( EDU.rutgers.stimulate.MIM.RawNLIEvent` **original** `)`

Methods

- *getSentence*
  `public Sentence getSentence( )`
- *getDuration*
  `public long getDuration( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.10 *Class* NLIStringEvent

Natural Language Input event, presented in String format.

Declaration

public class NLIStringEvent
extends EDU.rutgers.stimulate.MIM.MIEvent

Serializable Fields

- private String sentence
  –
- private long duration
  –

Constructors

- *NLIStringEvent*
  `public NLIStringEvent( java.awt.Component` **source**`, long` **when**`, long` **duration**`,`
  `java.lang.String` **sentence** `)`

Methods

- *getSentence*
  `public String getSentence( )`
- *getDuration*
  `public long getDuration( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.11 *Class* RawNLIEvent

provides the raw NLI input.

Declaration

public class RawNLIEvent
extends EDU.rutgers.stimulate.MIM.MIEvent

Serializable Fields

- private Sentence data
  –
- private long duration
  –

Constructors

- *RawNLIEvent*
  `public RawNLIEvent( java.awt.Component` **source**`, long` **when**`, long` **duration**`,`
  `EDU.rutgers.stimulate.MIM.SpeechParsing.Sentence` **sentence** `)`
  – Usage
    * Contructs a new RawNLIEvent. You'll have to know the pretty format.
  – Parameters
    * `source` - the source component.
    * `when` - the timestamp.
    * `sentence` - the sentence data.

Methods

- *getSentence*
  `public Sentence getSentence( )`
- *getDuration*
  `public long getDuration( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.12 *Class* RawPointer2DEvent

Event class for raw 2D pointer events.

Declaration

public class RawPointer2DEvent
extends EDU.rutgers.stimulate.MIM.MIEvent

Serializable Fields

- private final double X
  –
- private final double Y
  –

Constructors

- *RawPointer2DEvent*
  `public RawPointer2DEvent( java.awt.Component` **source**`, long` **when**`, double` **x**`, double` **y** `)`
  - Usage
    * Normal constructor, with the coordinates.
  - Parameters
    * `obj` - the source object.
    * `x,y` - the point where the event occurred.

Methods

- *getPoint*
  `public Point getPoint( )`
- *getX*
  `public double getX( )`
- *getY*
  `public double getY( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`

2.2.13 *Class* SimplePointer3DEvent

Event class for 3D position-only pointer events. It extends RawPointer2DEvent so you can retrieve the mapped 2D position in workspace. keep in mind that the 3D X and Y values are usually NOT the same as the 2D X and Y values!

Declaration

public class SimplePointer3DEvent
extends EDU.rutgers.stimulate.MIM.RawPointer2DEvent

Serializable Fields

- private Point3d point3d
  –
- private final Point3d originalPoint
  –

Constructors

- *SimplePointer3DEvent*
  `public SimplePointer3DEvent( java.awt.Component` **source**`, long` **when**`, double` **x**`, double` **y**`, javax.vecmath.Point3d` **point** `)`
  – Usage
    * Normal constructor, with the coordinates.
  – Parameters
    * `source` - the source object.
    * `when` - the timestamp.
    * `x,y` - the point where the event occurred.

Methods

- *getPoint3D*
  `public Point3d getPoint3D( )`
  - Usage
    - Get the 3D position of this event.
  - Returns - Point3d the 3D point where the event occurred.
- *getOriginalPoint3D*
  `public Point3d getOriginalPoint3D( )`
- *adjustPoint*
  `public void adjustPoint( javax.vecmath.Point3d` **newPoint** `)`

Methods inherited from class `EDU.rutgers.stimulate.MIM.RawPointer2DEvent`

(in 2.2.12, page 43)

- `public Point getPoint( )`
- `public double getX( )`
- `public double getY( )`

Methods inherited from class `EDU.rutgers.stimulate.MIM.MIEvent`

(in 2.2.6, page 37)

- `public long getWhen( )`

Methods inherited from class `java.util.EventObject`

- `public Object getSource( )`
- `public String toString( )`