<XML> A generic XML-structure for logging human computer interaction

Prepared by: Eric Van Horenbeeck, Tom Pauwaert,
Luuk Van Waes, Mariëlle Leijten

Version: 1.3

Version history: v 0.0 February 2012 - EvH, TP, LvW, ML - University of Antwerp
v 1.0 March 2012 - discussion with AW, JF - Lund University
v 1.1 May 2012 - discussion Cost conference in Poitiers
v 1.2 July 2012 - discussion Sig Writing conference in Porto
v 1.3 October 2015 - logging Chinese characters

Date: October 30, 2015

Project Members: **Scriptlog (Sweden)**
Asa Wengelin - Gothenborg University
Johan Frid - Lund University
Victoria Johansson - Lund University
Roger Johansson - Lund University
Inputlog (Belgium)
Luuk Van Waes - University of Antwerp
Mariëlle Leijten - Flanders Research Foundation/UA
Eric Van Horenbeeck - University of Antwerp
Tom Pauwaert - University of Antwerp
EyeWrite (UK)
Mark Torrance - Nottingham Trent University
Eye&Pen (France)
Denis Alamargot - University of Poitiers
HandSpy (Portugal)
Rui Alves - University of Porto

Disclaimer: this project has been prepared during the COST Action IS0703: European Research Network on Learning to Write Effectively (ERN-LWE) - working group 4 - Technological advances in writing tools.

A generic structure
for logging human computer interaction

This document provides a description of a generic XML log format for logging human computer interaction related to writing. It has been initiated by the developers of Scriptlog and Inputlog. In an initial stage both research groups have been discussing the best way to guarantee an optimal exchange of logging files between both programs. This resulted in a decision to develop an XML log standard that enables researchers to exchange log files between the two programs, utilizing the complementarity of the analyses that both programs offer. A further development of the XML log format to a more generic format, should create a standard for logging process information for a wider range of writing devices (e.g., digital pens, tablets, mobile devices).

The main aim of the project is to develop a **standardization** of the logging of digital writing processes. A generic XML structure (a) simplifies the interchangeability of research data, (b) the description of writing logging data, and (c) further establishes process logging as a research method.

Firstly, this format simplifies the ***interchangeability of research data*** collected with different logging tools. This enables researchers to exchange process data of various logging tools for further analysis. For instance, a logging session recorded by Scriptlog can therefore be analyzed via Inputlog. One of the main advantages is that the complementary functions of the logging tools are better exploited: you can run an Inputlog revision analysis on Scriptlog data, and you can run a word-in-context analyses on Inputlog data via the analyses module of Scriptlog. In other words, we strive to make data of the main logging tools interchangeable. At this moment Scriptlog, Inputlog, EyeWrite, Eye&Pen and HandSpy are involved in the project.

Secondly, the generic XML structure ***describes process data*** in a uniform and unambiguous way. We create a common perspective for the description of writing processes using different input devices (e.g. keyboard, speech, touchpad, digital pen) defined by a few generic primitives.

Thirdly, a more standardized description of logging data is also important to further ***establish logging as a main stream writing research method***. A generic structure for logging tools is certainly an important way to reach this goal. Complementarily, it is important to share expertise related to logging and analyzing writing processes on WritingPro: knowledge center for writing process research: www.writingpro.eu) is another.

The first part of the report describes the different components and how they are connected. Then an overview of this framework is presented in a scheme. The second part describes how the logged elements are represented in a XML notation.

Three main components

The log files consist of three main components.

1. Header

The header contains of:

- Meta data pertaining to different items not specifically related to a logging session, such as the version of the program used to log, the version of the program used to read and analyze the log, the processor used, a warning concerning the expected timing accuracy, and possibly others.
- Session data uniquely related to this log file, such as the identification of the person, the language used in the interaction, a timestamp.

The header has one essential condition: a timestamp that indicates the start of the logging process.

Users are free to add other elements as required by their specific logging program. As an example: Inputlog needs to know the language used in the text production to perform linguistic analysis. The format used is: a meta or session element name followed by a series of key/value pairs that describe the feature.

In the case of merging two or more log files, the session data (or a selection thereof) will be used to calculate a hash tag allowing to identify in the new merged log file the original file for every event. In the context of input logging the hash tag is a short piece of machine readable metadata added to an event and pointing to a bundle of session facts that are sufficient to trace back the original log file if necessary.

2. Footer

The footer contains of:

- Information that is only available after a session. E.g. the product statistics of a writing job (number of words, sentences, etc.), the average processor load over the session.
- Pointers to specific modules that may be used in an automated analysis process such as different processing steps that need to be executed without further human intervention.

A footer has no prerequisites. Researchers are free to add the elements required by their logging program in the format: module name followed by a series of key/value pairs that describe the feature.

3. Logged events

An 'event' is the top level entity that holds all the information concerning action(s) and its related parts. Although in most cases one event records one action, it will be possible to have many-action events. This happens when the subject is performing simultaneous tasks (e.g. typing and eye tracking) or when a particular analysis task demands the merging of some logging files on event level.

The central idea behind the structure of event logging is the distinction between:

- action
- properties of an action
- output

- properties of an output
- navigation & selection
- context of the logging session

The next page provides an overview of the proposed XML-structure.

Schematic overview XML structure

```
Header    meta
          ↳ entry
                ↳ key
                ↳ value

          session
          ↳ entry
                ↳ key

                ↳ value

Footer    module name = " "
          ↳ entry

                ↳ key
                ↳ value
```

Event	Primitive	Input type	Properties						
		optional param.	multi	x	y	seqnce	state	start T	end T
action	click (C)								
action	move (M)								
action	placeholder(P)								
action	sound (S)								
Use Cases									
pen movement	M	pen	x	√	√	√	pen down/ keyboard state	√	√
touch press	C	touch	√	√	√	x	?	√	√
swipe X	M	touch	x	√	√	√	x	√	√
left mouse button	C	mouse - button	x	√	√	x		√	√
scroll drag & drop	M	mouse - scroll dir	x	√	√	x	mouse state/ keyboard st	√	√
		- delta							
key press	C								
	P	work_break						√	√

Output type	Source	Value	Properties				
			pos	pos	file	value	
text	speech	string	x	x	sound file	path	
text	keyboard	string	√	√	position	int	
text	keyboard	string	√	√	text_area	int	

Focus type	Target	Properties		
window	browser			
text	the selected text	position		

Message	
system message	string (e.g. error condition)
other message	string (e.g. instruction given to the user)

A. Action

An action is every engagement of a person with a digital device, where this engagement needs to be recorded.

It is up to the specific logging tool to define what the minimal (atomic) action is. For a key logger this may be a mouse click or a key press, for an eye tracking device it may be the gaze on a screen, for a tablet the pressure of a stylus on the touch screen.

The proposed action taxonomy has four primitives and an open number of input types derived from these primitives.

Primitives:

- Click: a click is every action that can be described as a point in time and in space.
- Move: a move is every action with a certain duration in time and with one or multiple contiguous paths in space.
- Placeholder: a placeholder is an empty action that either is a point or a duration in time/space and provides contextual information about the passing of time of subsequent events. It may be ignored in the analysis.
- Other: special constraints related to other sensory information such as sound, smell or taste may justify using an additional action primitive.

Click, Move, and Other are abstract expressions for the real action instruments. These action instruments are defined as input types. The placeholder as such is not an instrument but it can be typed and provided with properties to supply additional information on the flow of events.

Input types:

Tightly linked to a specific technology, it is up to the researcher to provide the input types relevant for his/her software configuration. A placeholder action can be a work break or a session break. The placeholder action can also appear as the result of a post processing procedure. For instance the researcher may decide to ignore different focus changes to and from external sources and replace them with a placeholder.

Examples:

Primitive	*Input type - Use case*	*Configuration - Use case*
click	touch (tap)	touch pad on a laptop
click	key (press)	keyboard of a desktop
click	mouse (click)	desktop
move	swipe (X)	touch screen of a smartphone
move	mouse (drag & drop)	laptop
move	mouse (scroll)	desktop
move	stylus (write)	touch screen of a tablet
placeholder	merging	fusion point of two ifdxs
placeholder	work break	user takes a break
other	sound	instruction to the user

Optional parameters:

Each input type can be extended with additional functionality.

Examples:

Primitive	*Input type- Use case*	*Parameters - Use case*
click	touch (tap)	pressure
click	key (press)	position, replay, doc length
click	mouse (click)	button left
move	swipe (X)	
move	mouse (drag & drop)	pause threshold
move	mouse scroll	orientation, delta
move	stylus (write)	orientation, pressure
placeholder	session break	url to original idfx
placeholder	work break	
other	sound	wav-file

B. Properties of an action

For an action to be recordable, certain hardware and software provisions need to be present.

The program needs to log *position* and *timing* to record the moments in space and time of an event. A third element to record is the *state* of certain components. The state is used to give a description of the behavior of a component in response to explicit events. For instance, when logging a writing process on a standard computer a keyboard event might have the shift key locked (VK_SHIFTLOCK) which in combination with the 'f' key (VK_F) produces the capital letter 'F'. With knowledge of the state of the shift key (state: key down) the program knows that the next letter will be a capital. Note: the key up of this action can be derived from the timing properties.

A single point in space is defined by its X/Y coordinates; an instant in time by its start and end moments. A series of consecutive X/Y points on a space is *sequence* (a path), and a *duration* is the time between start and end. The duration is not recorded in the log but is calculated in the analysis as the difference between start and end. A placeholder may contain position and/or timing information.

Examples:

				Position		*Timing*			
Primitive	*Type*	*Multiplicity*	*Sequence*	*X*	*Y*	*Start*	*End*	*State*	*Properties*
click	Mouse click	No	No	√	√	√	√	key shift down	Button
move	Multi touch swipe	Yes*	Yes	√	√	√	√		
click	Eye track gaze	Yes**	Yes	√	√	√	√		
placeholder	Transition	No	Yes	√	√	√	√		

* Five-finger gesture on an iPad
** Each eye has its own X/Y coordinates

Finally, there is the *multiplicity* property that indicates whether a given position occurs more than once in the same action. This happens e.g. in eye tracking where the position of the left and the right eye are recorded separately, both, however, are in the same action. As with duration, multiplicity needs not to be recorded but is implied from the existence of multiple positions and will be used accordingly in the analysis of the data.

C. Output

The output of an action is every recordable effect of an action.

Outputs are typed and have a source. The output value is the actual result of an action as it is presented onto some form of hardware.

Examples:

Device	*Output type*	*Source*	*Value*
keyboard	text	keyboard	"b"
touch screen	text	virtual keyboard	"v"
smartphone	text	speech	"This is a sentence I have spoken and that is transcribed"

D. Properties of an output

Depending on the specific device used in the logging process and depending on the recording conditions, certain properties can be used to further describe the output. An example of such a property is the 'text_area'. This allows the logging program to define complex documents with multiple input regions. Another property could be 'style', describing the formatting of a text.

Examples:

Device	*Output type*	*Source*	*Value*	*Property*	*Property value*
smartphone	text	speech	"This is a sentence I have spoken and that is transcribed"	sound file	C:/some_path/ your_sentence.wav
keyboard	text	keyboard	"This is pasted text"	text_area	#2
keyboard	text	keyboard	"This is pasted text"	position	42
keyboard	text	keyboard	"This is pasted text"	style	bold

E. Navigation & Selection

We are now left with a series of secondary feats that shape the output, but that are not visible in the final product. They involve the navigation in and around the document, and selections of (text) output; the latter will typically be copied, replaced, deleted, or inserted.

The nature of the selections is signaled by a change of focus that is recorded and typed. The target of the focus change describes the new content in view, e.g. a selection of text with a mouse action. Additional properties may complement the focus description such as the start and the end position of the selected text.

Examples:

Focus type	*Target*	*Property*
window	IExplorer - http://www.ua.ac.be	previousWindowFocusTime
text	The selected text	start position, end position

An *insertion* is a keyboard action, such as the pressing of CTRL+V that copies text onto the screen. This is expressed by having an output event of the type 'text', where the pasted text is the value of the output. The position is logged as a property of the output element.

A *deletion* of a selection could be the pressing of a backspace or delete key after having selected some text. This would be expressed by having a focus change of the type 'text', where the selection is the value of the focus change together with the beginning and end positions of the selection as properties. The output of this event would then be an output element with an empty value, describing that there has been output, but that the output is empty.

A *replacement* is very much like a deletion, the difference being that the output in this case would not be empty but would be a single character or a string of characters.

F. Context of the logging session

This category allows the researcher or the computer system to include extra information that describes relevant situations that may have influenced the logged data or the logging process. It takes the form of a *message* added to the event. A system message could be an error condition that occurred at a certain point. A message could report the end of a session and is included in the session break placeholder action.

XML notation example - (Handmade example, not an actual logging file)

```
<?xml version="1.0" encoding="utf-8"?>
<log>
        <!--- Header -->
        <meta>
                <entry   key ="__LogProgramVersion">  5.0.1.11  </entry>
                <entry key ="__LogCreationDate">   15/11/11 11:45:07.737 </entry>
        </meta>
         <session hash = "#13321478">
                <entry key ="Participant"> Bob </entry>
                <entry key ="Language"> EN </entry>
        </session>

        <!-- Body -->
        <!-- First click and select some text, the clicking is not represented here in the XML -->
        <event  id = "0"  hash = "#13321478">
                <action type = "click"  inputtype = "mouse">
                                <param name = "button" > LBUTTON </param>
                                <state type = "keyboard">
                                        <entry key = "key_down"> VK_SHIFT</entry>
                                </state>
                                <position>
                                         <x> ... </x>
                                         <y> .... </y>
                                </position>
                                <timing>
                                        <start> ... </start>
                                        <end> ... </end>
                                </timing>
                                <message>error #560</message>
                </action>
        </event>
        <event  id = "1"  hash = "#13321478">
                <action type = "click"  inputtype = "keyboard">
                        <timing>
                                <start>193061736</start >
                                <end>193061798</end>
                        </timing>
                        <param name="key">VK_L</param>
                        <param name="value">l</param>
                        <param name="documentLength">22</param>
                        <param name="position">21</param>
                        <param name="replay">True</param>
                </action>
                <output type="text" source="keyboard">
                        <value>l</value>
                </output>
        </event>

        <!--  Replacement example  -->
        <event  id = "2"  hash = "#13321478">
                <action type = "move"  inputtype = "mouse">
                        <timing>
                                <start>193061826</start >
                                <end>193061978</end>
                        </timing>
                        <param name = "button" > LBUTTON </param>
                        <param name="documentLength">23</param>
                        <param name="replay">False</param>
                        <position>
                                <pos>
                                        <x>20</x>
                                        <y>115</y>
                                </pos>
                                <pos>
                                        <x>31</x>
                                        <y>115</y>
                                </pos>
                                <pos>
                                        <x>42</x>
                                        <y>115</y>
                                </pos>
                        </position>
                </action>
                <focus type="text">
                        <target>I am selected text</target>
                        <property name="begin-position">0</property>
                        <property name="end-position">18</property>
                </focus>
        </event>

        <!-- Start paste action -->
        <event  id = "3"  hash = "#13321478">
                <action type = "click"  inputtype = "keyboard">
                        <timing>
                                <start>193061836</start >
                                <end>193061998</end>
                        </timing>
                        <param name="key">VK_LCTRL</param>
                        <param name="value" />
                        <param name="position">0</param>
                        <param name="documentLength">23</param>
                        <param name="replay">False</param>
                        <state type = "keyboard">
                                <entry key = "key_down"> VK_LCTRL</entry>
                         </state>
                </action>
                <focus type="text">
                        <target>I am selected text</target>
                        <property name="begin-position">0</property>
                        <property name="end-position">18</property>
                </focus>
        </event>

        <!-- Paste text -->
        <event  id = "4"  hash = "#13321478">
                <action type = "click"  inputtype = "keyboard">
                        <timing>
                                <start>193072136</start >
                                <end>193072598</end>
                        </timing>
                        <param name="key">VK_V</param>
                        <param name="value" />
```

```
                    <param name="position">0</param>
                    <param name="documentLength">23</param>
                    <param name="replay">False</param>
                    <state type = "keyboard">
                              <entry key = "key_down"> VK_LCTRL</entry>
                     </state>
          </action>
          <focus type="text">
                    <target>I am selected text</target>
                    <property name="begin-position">0</property>
                    <property name="end-position">18</property>
          </focus>
          <output type="text" source="keyboard">
                    <value>I am pasted</value>
          </output>
</event>

<event  id = "5"  hash = "#13321478">
          <action type="click" inputtype="mouse">
                    <timing>
                              <start>193172136</start >
                              <end>193182598</end>
                    </timing>
                    <position>
                              <x>200</x>
                              <y>12</y>
                    </position>
                    <param name="button">LBUTTON</param>
                    <param name="documentLength">16</param>
                    <param name="replay">False</param>
          </action>

          <!-- Click in the text, cursor change but no selection -->
          <focus type="text">
                    <target />
                    <property name="begin-position">5</property>
                    <property name="end-position">5</property>
          </focus>
</event>

<!-- Session interrupted -->
<event  id = "6"  hash = "#13321478">
          <action type="placeholder" inputtype="work_break">
                    <timing>
                              <start>193272136</start >
                              <end>200282598</end>
                    </timing>
                    <message>coffee break</message>
          </action>
</event>

<!--- Footer -->
<module name = "statistics"  hash = "#13321478">
          <entry key ="charexclspaces"> 118 </entry>
          ...
          <entry key ="..." ></entry>
</module>
```

```
		<module name = "general analysis"  hash = "#13321478">
			<entry key ="" ></entry>
		</module>
		<module name = "pause analysis"  hash = "#13321478">
			<entry key ="pause threshold"> 2000 </entry>
			<entry key ="fixed interval"> 60 </entry>
		</module>
</log>
```

Logging for Simplified Chinese - Explained

When writing Chinese texts in Word the logging requirements are to a certain extent quite different from logging writing that uses the roman alphabet. The most commonly used method is called the Hanyu Pinyin Input Method for Mandarin Chinese and is the official system to transcribe Chinese characters into Latin script in the People's Republic of China, Taiwan, and Singapore. Pinyin is a Romanization system that transcribes the sounds of Mandarin using the western (Roman) alphabet. Pinyin is most commonly used in Mainland China. It is often used to teach Standard Chinese and spell Chinese names in foreign publications and is used as an input method to enter Chinese characters into computers.

"For instance 外匯買賣 (Foreign Exchange Trading) can be typed by Pinyin code as: wai hui mai mai. Different interfaces and systems exist side by side. They all have their strengths and weaknesses. For instance, the Sogou Pinyin Method (搜狗拼音输入法; Pinyin: Sōugǒu Pīnyīn Shūrùfǎ) is a popular Chinese Pinyin input method. By July 2011, Sogou Pinyin had an 83.6% penetration rate with more than 300 million users

For comparative analyses of writing processes in Western and Chinese languages, an adapted pause framework is needed to deal with the dynamics of writing that characterize text production. An integration of the logging of Chinese character input in a standardized XML structure would certainly facilitate interlingual research and collaboration.

An example:

1. Write in Pinyin: 'yifenzhong'.
2. Word offers a choice of different Chinese characters to use.
3. Press '1' to select the first choice.
4. Word replace 'yifenzhong' with the Simplified Chinese: '一分钟'.

We will now show how to log such behavior correctly, according to this standard. The examples we will present assume two possible, different implementations of the above behavior, provided by Word:

- The first possibility is that all characters in Pinyin are buffered and do not appear in the document. When the Simplified Chinese is selected from the popups provided by Word, the Chinese characters are inserted into the document.
- The other possibility is that as the characters are typed in Pinyin, they are simultaneously added to the Word document. While you type in Pinyin, Word continuously shows the popup window offering the choice from different Chinese characters for the Pinyin you just typed. Then, as you select the right Chinese characters from the popup dialog the Pinyin text is replaced by the Chinese characters.

The first approach we call the 'buffered' approach, and the second approach will be referred to as the 'unbuffered' approach.

Following we provide XML examples for both approaches:

Buffered approach

```
<?xmlversion="1.0" encoding="utf-8"?>
<log>
	<!-- Header -->
	<meta>
		...
	</meta>
	<session hash = "#123ABC">
		...
	</session>

	<!-- Body -->
	<!-- Characters are being 'buffered' -->
	<!-- Type yifenzhong and select the Chinese characters -->
	<event id="0" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
				<start>193061736</start>
				<end>193061833</end>
			</timing>
			<param name="key">VK_Y</param>
			<param name="value">y</param>
			<param name="documentLength">1</param>
			<param name="position">0</param>
			<param name="replay">False</param>
		</action>
	</event>

	<event id="1" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
				<start>193061950</start>
				<end>193062070</end>
			</timing>
			<param name="key">VK_I</param>
			<param name="value">i</param>
			<param name="documentLength">1</param>
			<param name="position">0</param>
			<param name="replay">False</param>
		</action>
	</event>
	...
	<!-- We have now typed up to "yifenzho", these are the last characters -->
	<event id="8" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
				<start>193073002</start>
				<end>193073150</end>
			</timing>
			<param name="key">VK_N</param>
			<param name="value">n</param>
			<param name="documentLength">1</param>
			<param name="position">0</param>
			<param name="replay">False</param>
```

```
        </action>
    </event>

    <event id="9" hash="#123ABC">
        <action type="click" inputtype="keyboard">
            <timing>
                <start>19307203</start>
                <end>193062339</end>
            </timing>
            <param name="key">VK_G</param>
            <param name="value">g</param>
            <param name="documentLength">1</param>
            <param name="position">0</param>
            <param name="replay">False</param>
        </action>
    </event>

    <!-- "yifenzhong" has been typed, select the right Chinese characters
     -- The Chinese characters are considered the output of the pressing
     -- the number 1, to select the first set of characters.
     -->
    <event id="10" hash="#123ABC">
        <action type="click" inputtype="keyboard">
            <timing>
                <start>19307400</start>
                <end>193062581</end>
            </timing>
            <param name="key">VK_1</param>
            <param name="value">1</param>
            <param name="documentLength">1</param>
            <param name="position">0</param>
            <param name="replay">False</param>
        </action>
        <output type="text" source="keyboard">
            <value>一分钟</value>
        </output>
    </event>

    <!-- Continue typing... -->
    <event id="11" hash="#123ABC">
        <action type="click" inputtype="keyboard">
            <timing>
                <start>19307592</start>
                <end>193062666</end>
            </timing>
            <param name="key">VK_Y</param>
            <param name="value">y</param>
            <param name="documentLength">4</param>
            <param name="position">3</param>
            <param name="replay">False</param>
        </action>
    </event>
    ...
</log>
```

Unbuffered approach

```
<?xmlversion="1.0" encoding="utf-8"?>
<log>
	<!-- Header -->
	<meta>
		...
	</meta>
	<session hash = "#123ABC">
		...
	</session>

	<!-- Body -->
	<!-- Characters are being 'buffered' -->
	<!-- Type yifenzhong and select the Chinese characters -->
	<event id="0" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
				<start>193061736</start>
				<end>193061833</end>
			</timing>
			<param name="key">VK_Y</param>
			<param name="value">y</param>
			<param name="documentLength">1</param>
			<param name="position">0</param>
			<param name="replay">True</param>
		</action>
		<output type="text" source="keyboard">
			<value>y</value>
		</output>
	</event>

	<event id="1" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
				<start>193061950</start>
				<end>193062070</end>
			</timing>
			<param name="key">VK_I</param>
			<param name="value">i</param>
			<param name="documentLength">2</param>
			<param name="position">1</param>
			<param name="replay">True</param>
		</action>
		<output type="text" source="keyboard">
			<value>i</value>
		</output>
	</event>

	...

	<!-- We have now typed up to "yifenzho", these are the last characters -->
	<event id="8" hash="#123ABC">
		<action type="click" inputtype="keyboard">
			<timing>
```

```
                    <start>193073002</start>
                    <end>193073150</end>
               </timing>
               <param name="key">VK_N</param>
               <param name="value">n</param>
               <param name="documentLength">9</param>
               <param name="position">8</param>
               <param name="replay">True</param>
          </action>
          <output type="text" source="keyboard">
               <value>n</value>
          </output>
     </event>

     <event id="9" hash="#123ABC">
          <action type="click" inputtype="keyboard">
               <timing>
                    <start>19307203</start>
                    <end>193062339</end>
               </timing>
               <param name="key">VK_G</param>
               <param name="value">g</param>
               <param name="documentLength">10</param>
               <param name="position">9</param>
               <param name="replay">True</param>
          </action>
          <output type="text" source="keyboard">
               <value>g</value>
          </output>
     </event>

     <!-- "yifenzhong" has been typed, select the right Chinese characters
       -- The Chinese characters are considered the output of the pressing
       -- the number 1, to select the first set of characters.
       -- The original text in Pinyin (yifenzhong) is _replaced_ by the output.
       -- So, the focus attribute of type text siginifies a text selection.
       -- The output of type text while text is being focused means the output
       -- will be replacing the currently selected text.
       -->
     <event id="10" hash="#123ABC">
          <action type="click" inputtype="keyboard">
               <timing>
                    <start>19307400</start>
                    <end>193062581</end>
               </timing>
               <param name="key">VK_1</param>
               <param name="value">1</param>
               <param name="documentLength">1</param>
               <param name="position">0</param>
               <param name="replay">False</param>
          </action>
          <focus type="text">
               <target>yifenzhong</target>
               <property name="begin-position">0</property>
```

```
                        <property name="end-position">10</property>
                </focus>
                <output type="text" source="keyboard">
                        <value>一分钟</value>
                </output>
        </event>

        <!-- Continue typing... -->
        <event id="11" hash="#123ABC">
                <action type="click" inputtype="keyboard">
                        <timing>
                                <start>19307592</start>
                                <end>193062666</end>
                        </timing>
                        <param name="key">VK_Y</param>
                        <param name="value">y</param>
                        <param name="documentLength">4</param>
                        <param name="position">3</param>
                        <param name="replay">False</param>
                </action>
                <output type="text" source="keyboard">
                        <value>y</value>
                </output>
        </event>
        ...
</log>
```

Clicking chinese characters as opposed to using number selection
When you click on the right sequence of chinese characters in the popup window instead of using a number to select the sequence, the above output would use a moues action with output instead. This is shown below:

```
<event id="10" hash="#123ABC">
                <action type="click" inputtype="mouse">
                        <timing>
                                <start>19307400</start>
                                <end>193062581</end>
                        </timing>
                        <position>
                                <x>200</x>
                                <y>12</y>
                        </position>
                        <param name="button">LBUTTON</param>
                        <param name="documentLength">10</param>
                        <param name="replay">False</param>
                </action>
                <focus type="text">
                        <target>yifenzhong</target>
                        <property name="begin-position">0</property>
                        <property name="end-position">10</property>
                </focus>
                <output type="text" source="keyboard">
                        <value>一分钟</value>
                </output>
        </event>
```