

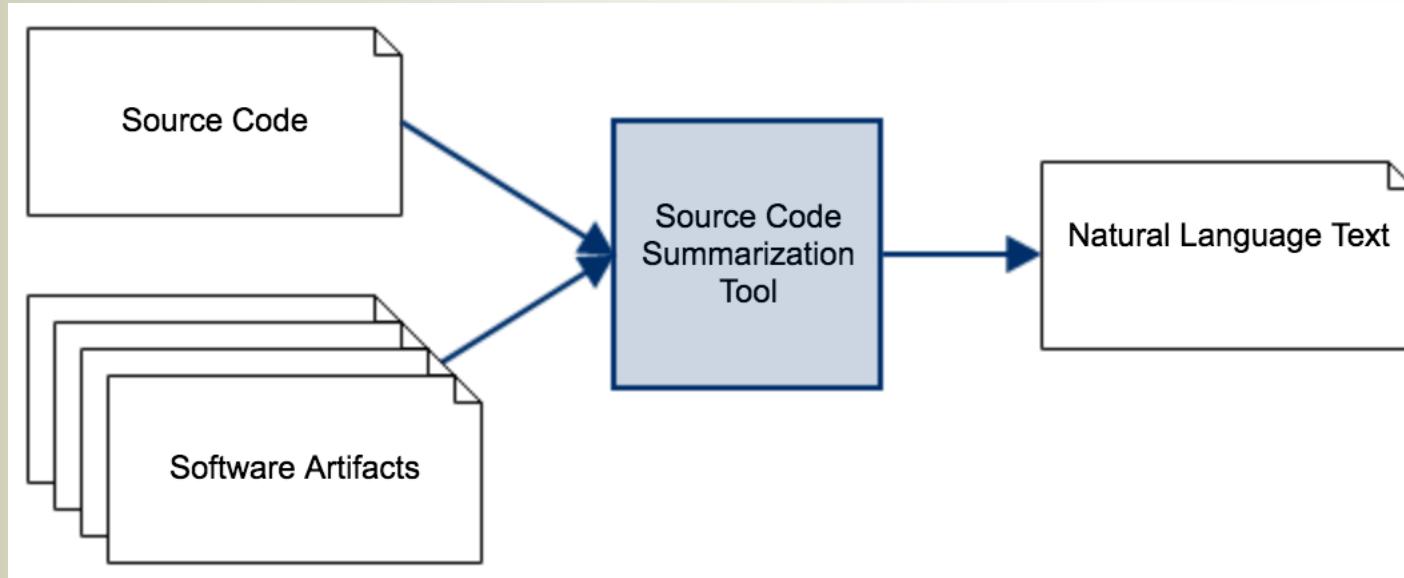
Summary

- Reproducibility package for experiments in source code summarization
- Conversion of two techniques for source code summarization to be used in TraceLab
- Creation of a set of components for the TraceLab infrastructure

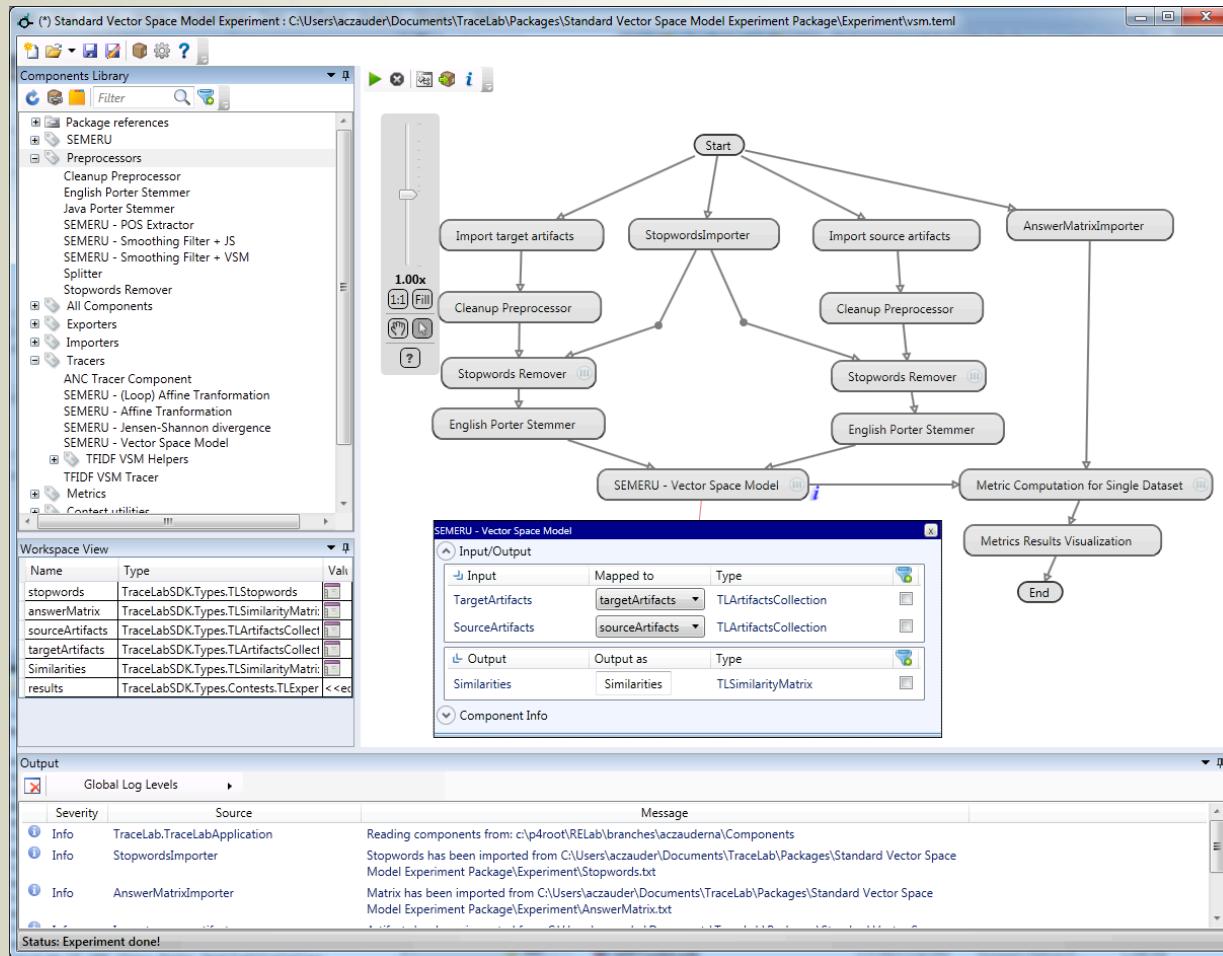


Source Code Summarization

- The process of creating natural language descriptions of source code



TraceLab



<http://coest.org/index.php/tracelab/tracelab-download>

Main Output

- Ported two techniques for source code summarization to TraceLab
 - Abstractive
 - Extractive



Extractive Technique

- Rodeghero et al. – An Eye-Tracking Study of Java Programmers and Application to Source Code Summarization
- Higher scores to common key words particular to a method, but rare through the source code

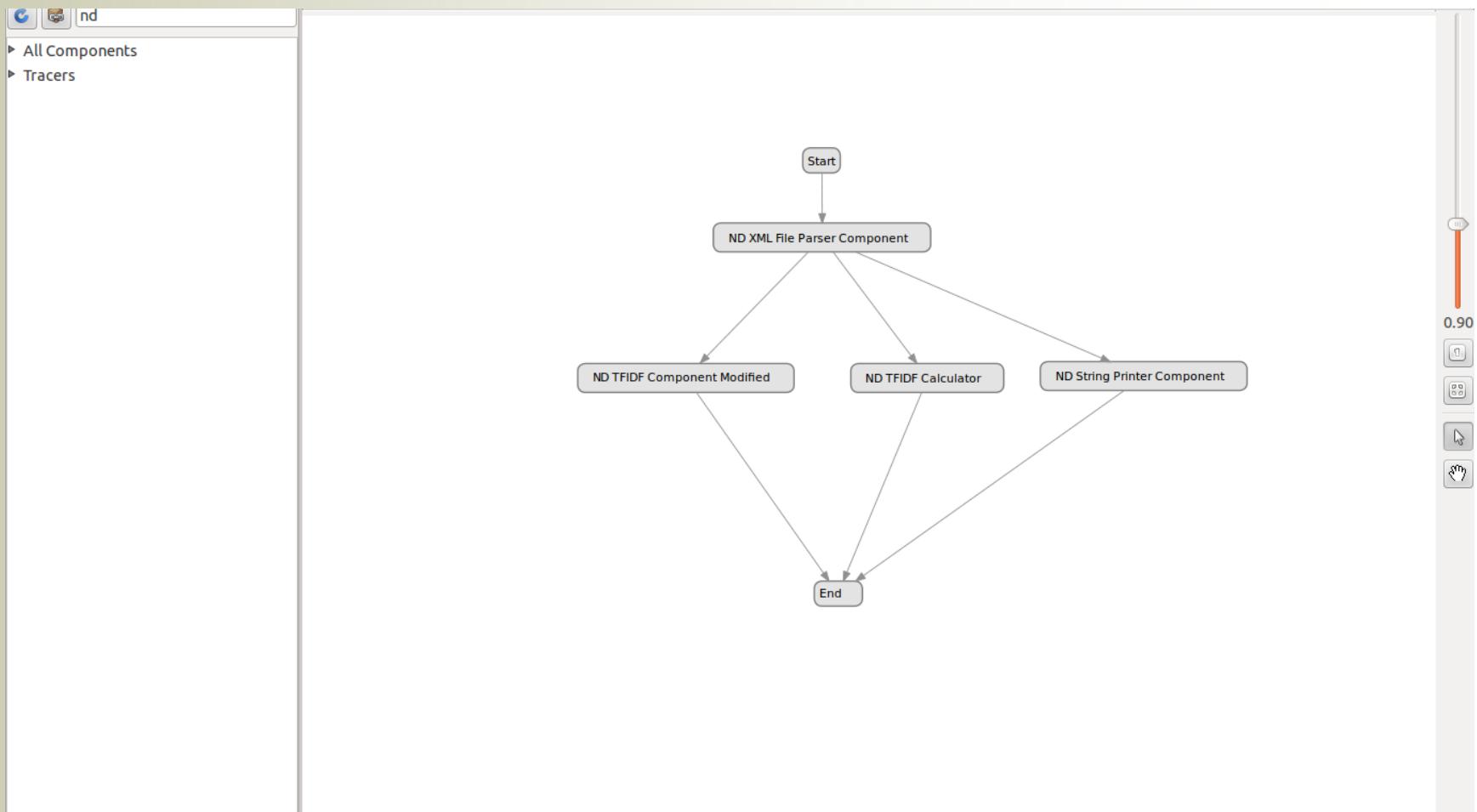


Extractive Technique

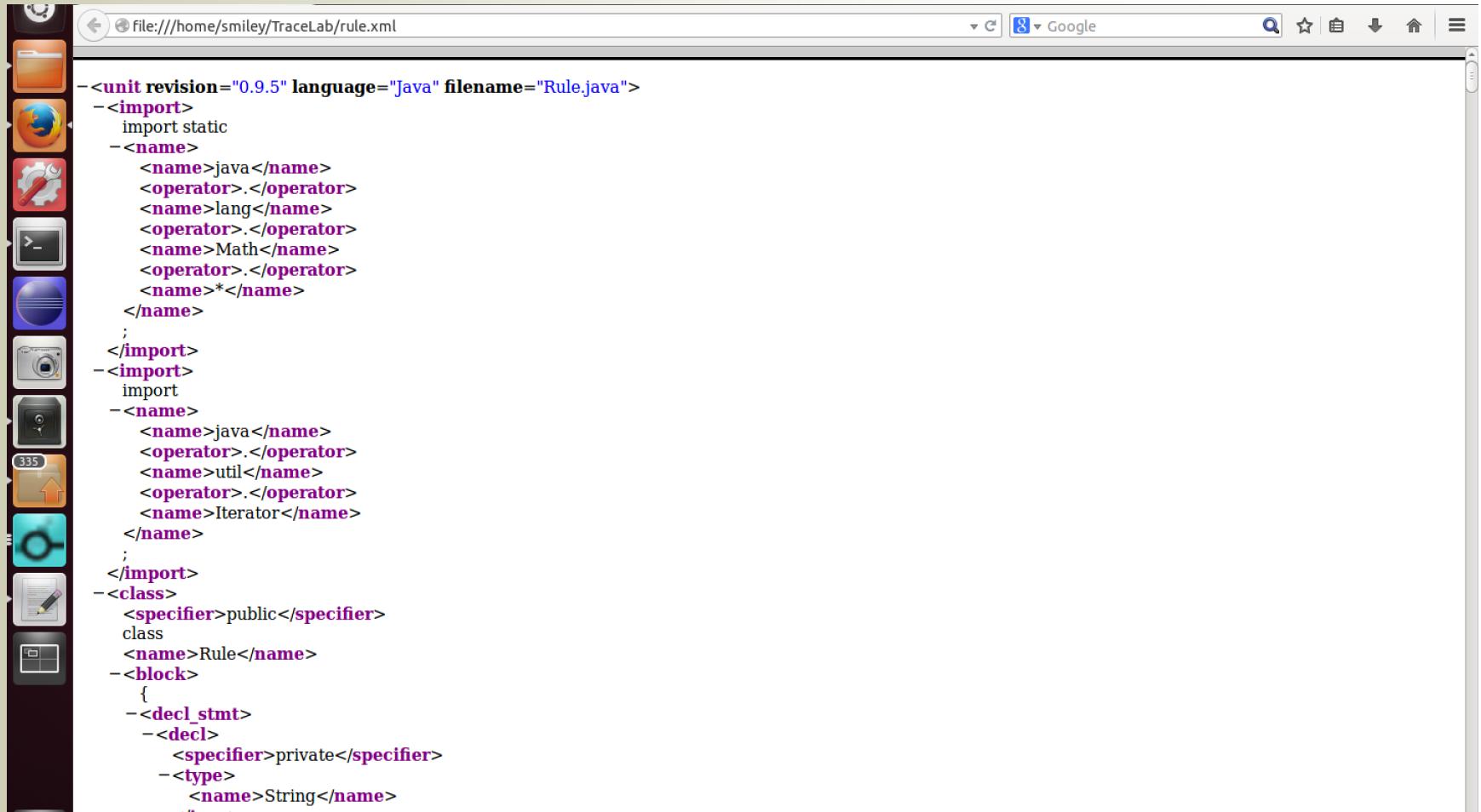
- Follow the previously described procedure
- Main modifications
 - Different weights to where the keywords are located in the source code
 - Method Signature
 - Control flow
 - Method Invocation



Modified VSM TF/IDF



Input



The screenshot shows a Java code editor window with a sidebar containing various icons. The main pane displays the following XML-like code:

```
-<unit revision="0.9.5" language="Java" filename="Rule.java">
-<import>
    import static
-<name>
    <name>java</name>
    <operator>.;</operator>
    <name>lang</name>
    <operator>.;</operator>
    <name>Math</name>
    <operator>.;</operator>
    <name>*</name>
</name>
;
</import>
-<import>
    import
-<name>
    <name>java</name>
    <operator>.;</operator>
    <name>util</name>
    <operator>.;</operator>
    <name>Iterator</name>
</name>
;
</import>
-<class>
    <specifier>public</specifier>
    class
    <name>Rule</name>
    -<block>
        {
        -<decl_stmt>
            -<decl>
                <specifier>private</specifier>
                -<type>
                    <name>String</name>
                </type>
```

Output

```
tfidf_mod_calculate...6-10-04 22-54-33.txt ✘
Rule-String-inklass-String-incolumn-String-infeature
[score, condition, new, conditions, klass, infeature, incolumn, inklass, string, rule]
[0.0924, 0.1848, 0.2584, 0.1307, 0.0924, 0.3696, 0.3696, 0.2772, 0.2326, 0.0924]

Rule-String-inklass
[score, condition, new, conditions, klass, inklass, string, rule]
[0.1848, 0.1848, 0.2584, 0.1307, 0.1848, 0.5545, 0.155, 0.1848]

getKlass
[return, getklass, klass]
[0.2772, 0.6695, 0.5545]

getScore
[getscore, return, score]
[1.109, 0.2772, 0.5545]

getMaxScore
[maxscore, getmaxscore, return]
[0.6695, 1.109, 0.2772]

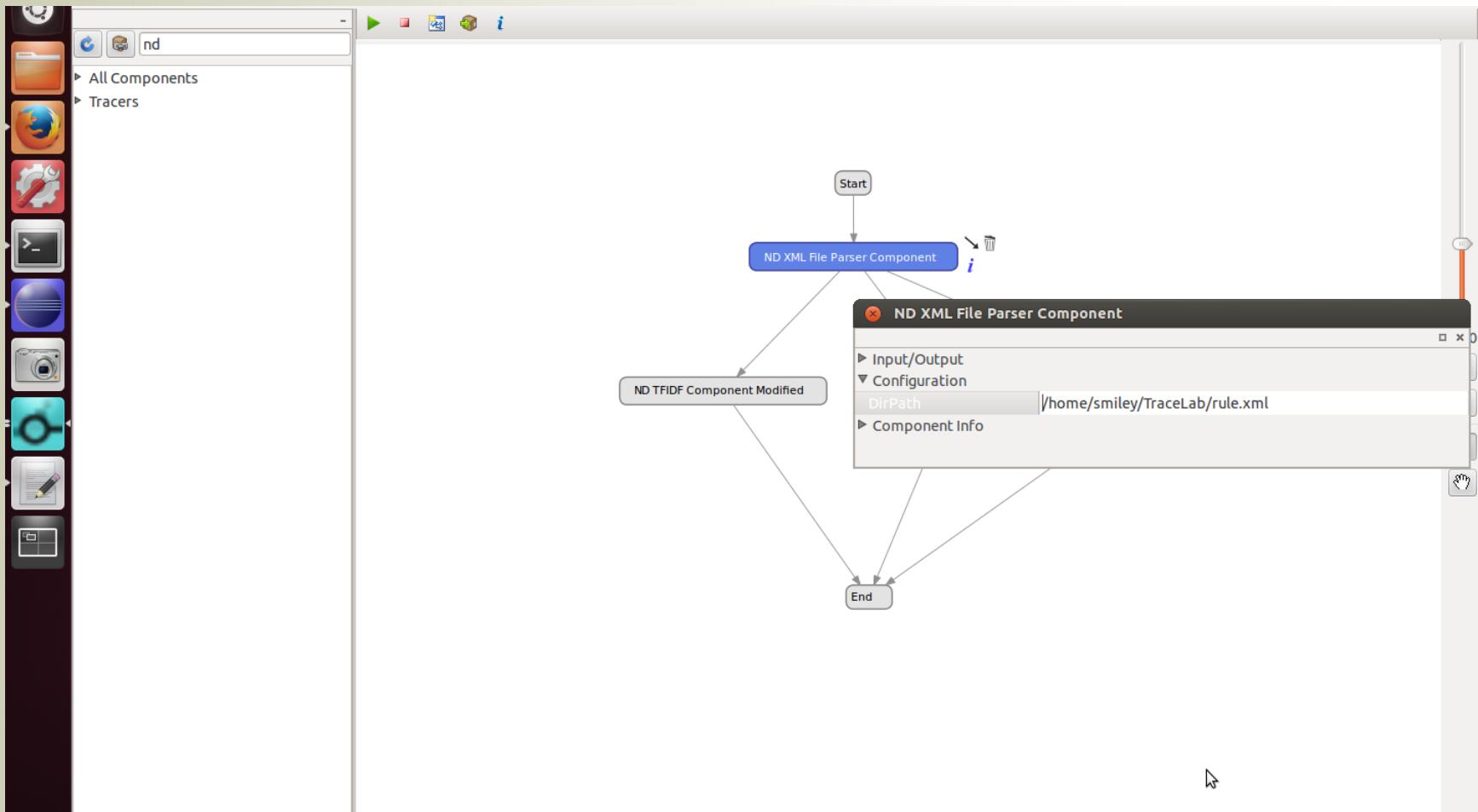
resetMaxScore
[resetmaxscore, maxscore]
[1.6635, 1.0043]

getScoreSum
[scoresum, getscoresum, return]
[0.6695, 1.109, 0.2772]

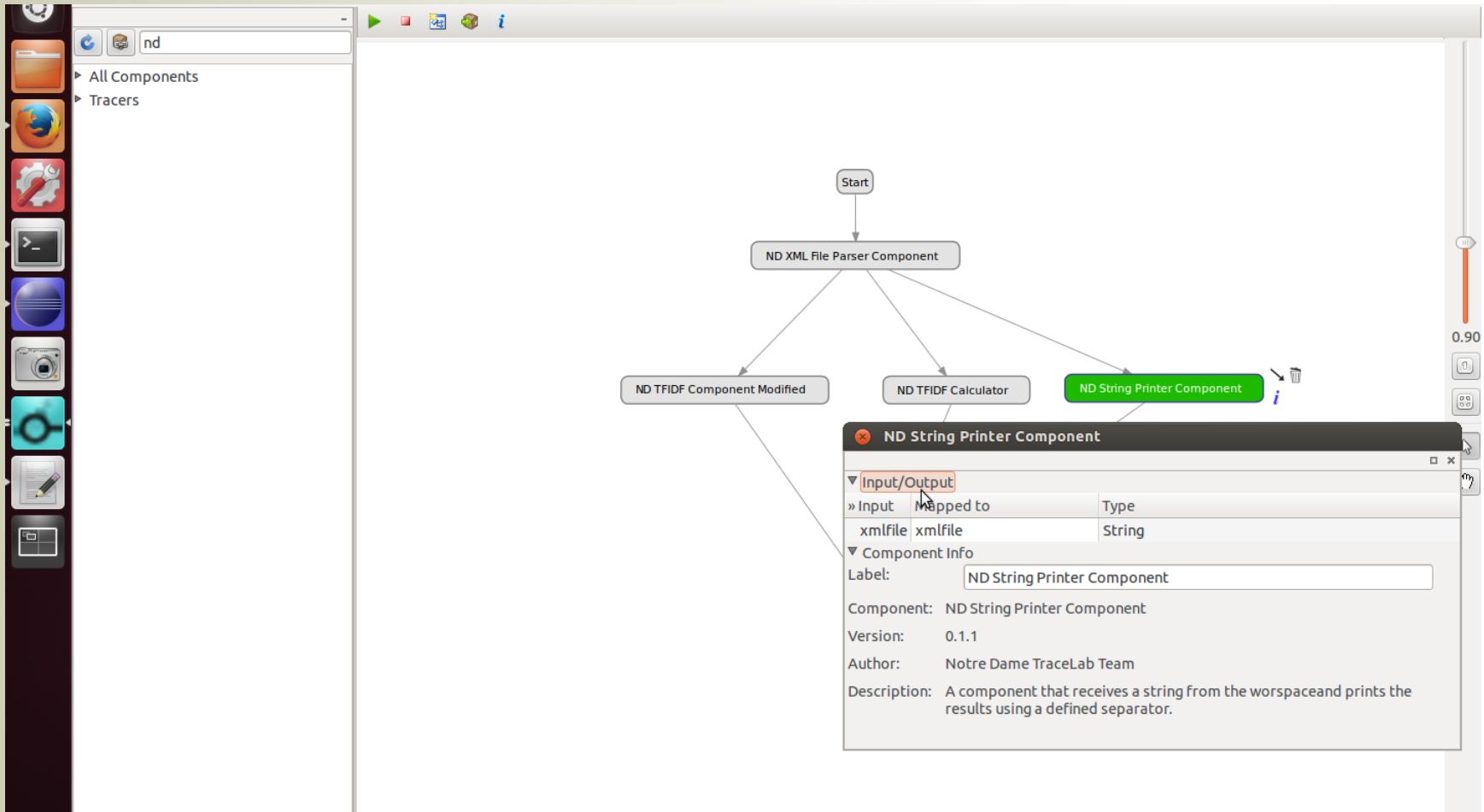
resetScoreSum
[resetscoresum, scoresum]
[1.6635, 1.0043]

hasImproved
[else, true, false, newmax, if, hasimproved, return, rule]
```

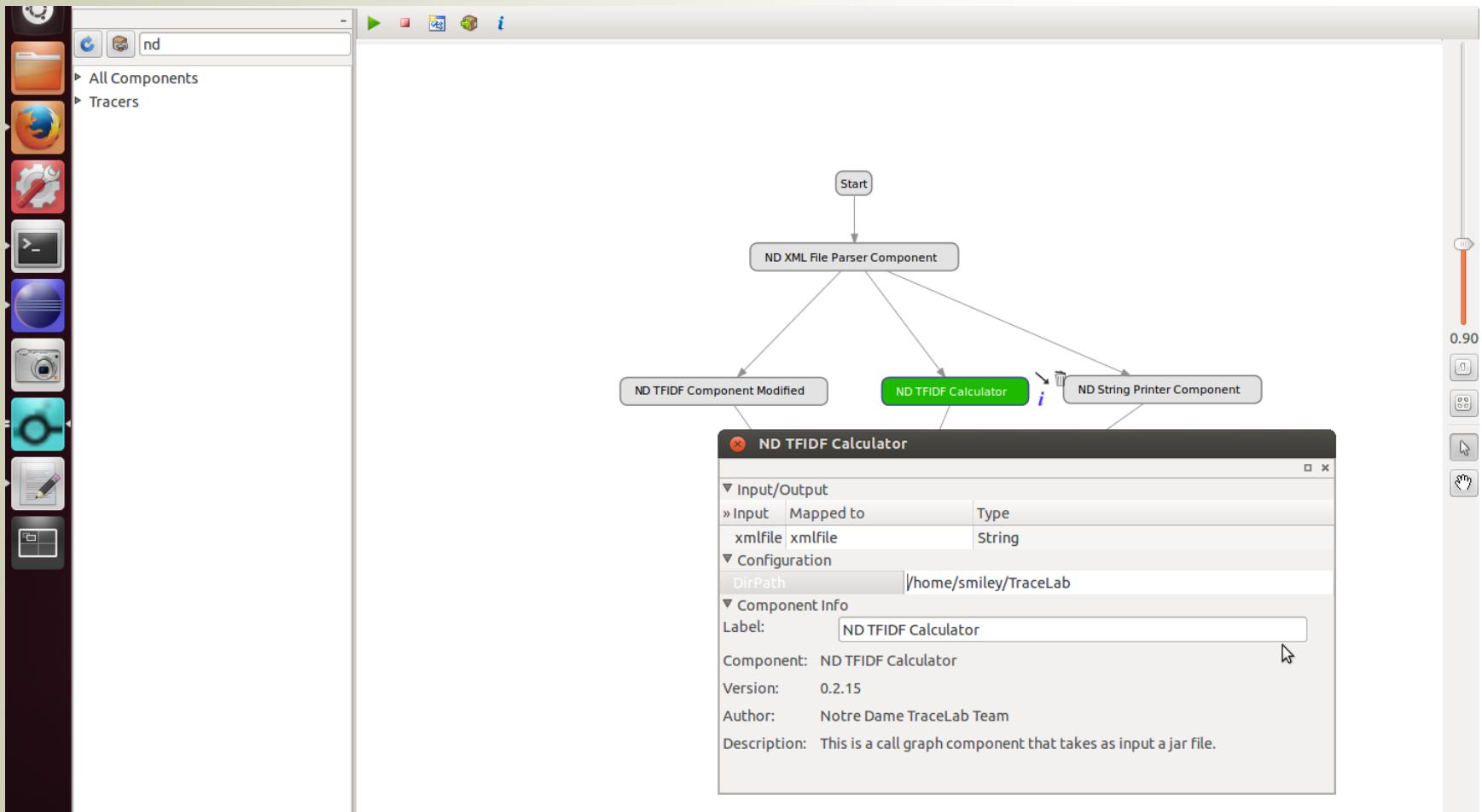
ND XML File Parser Component



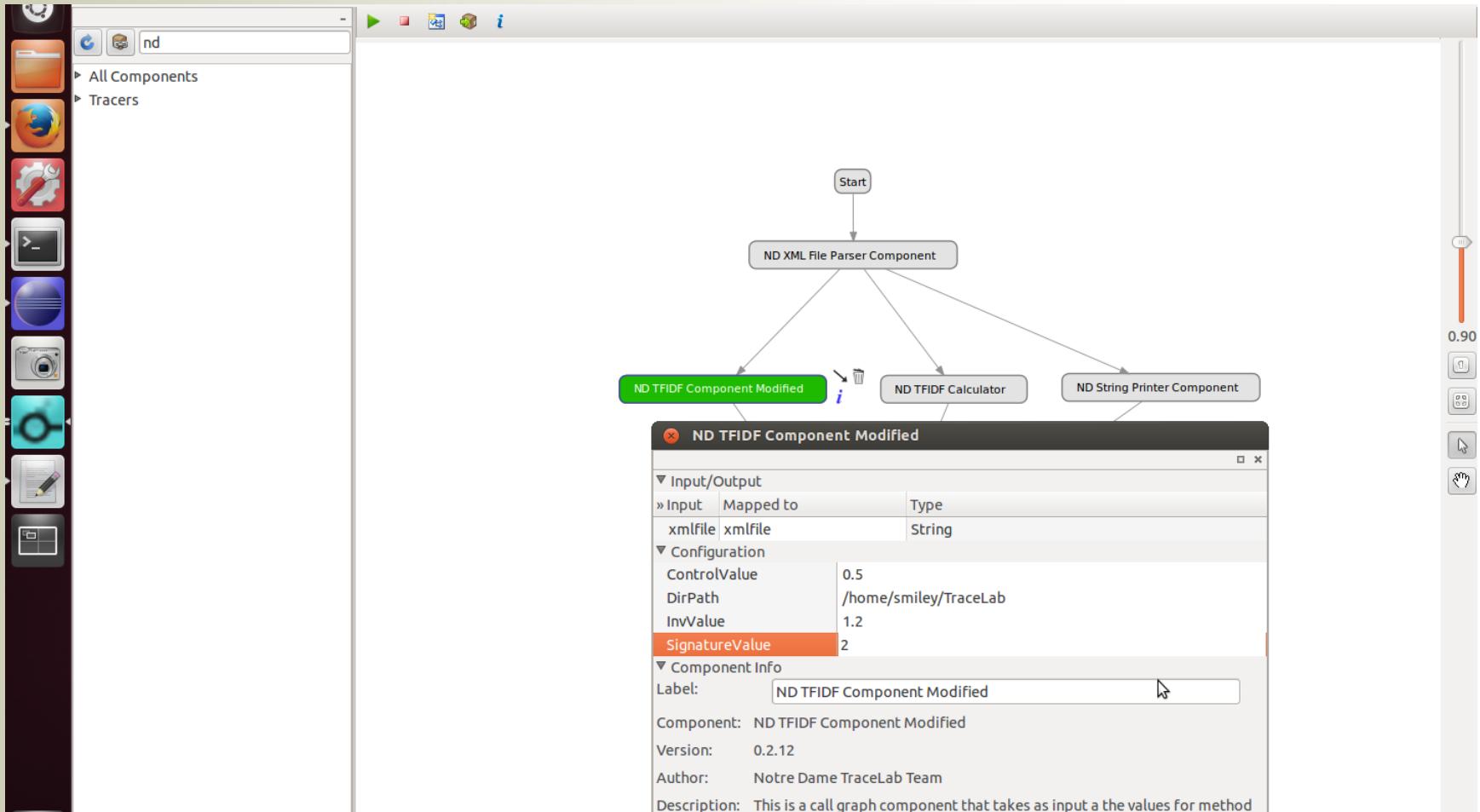
ND String Printer Component



ND TF/IDF Calculator



ND TF/IDF Component Modified



Abstractive Technique

- McBurney et al. Automatic Source Code Summarization of Context for Java Methods
- Provide high-level summaries of a method and its context
- Step-by-Step execution:
 - Collect contextual data about Java methods



Abstractive Technique

- McBurney et al. Automatic Source Code Summarization of Context for Java Methods
- Provide high-level summaries of a method and its context
- Step-by-Step execution:
 - Collect contextual data about Java methods
 - Use key words to describe how a method is used



Abstractive Technique

- McBurney et al. Automatic Source Code Summarization of Context for Java Methods
- Provide high-level summaries of a method and its context
- Step-by-Step execution:
 - Collect contextual data about Java methods
 - Use key words to describe how a method is used
 - Use SWUM to identify parts-of-speech for the different keywords



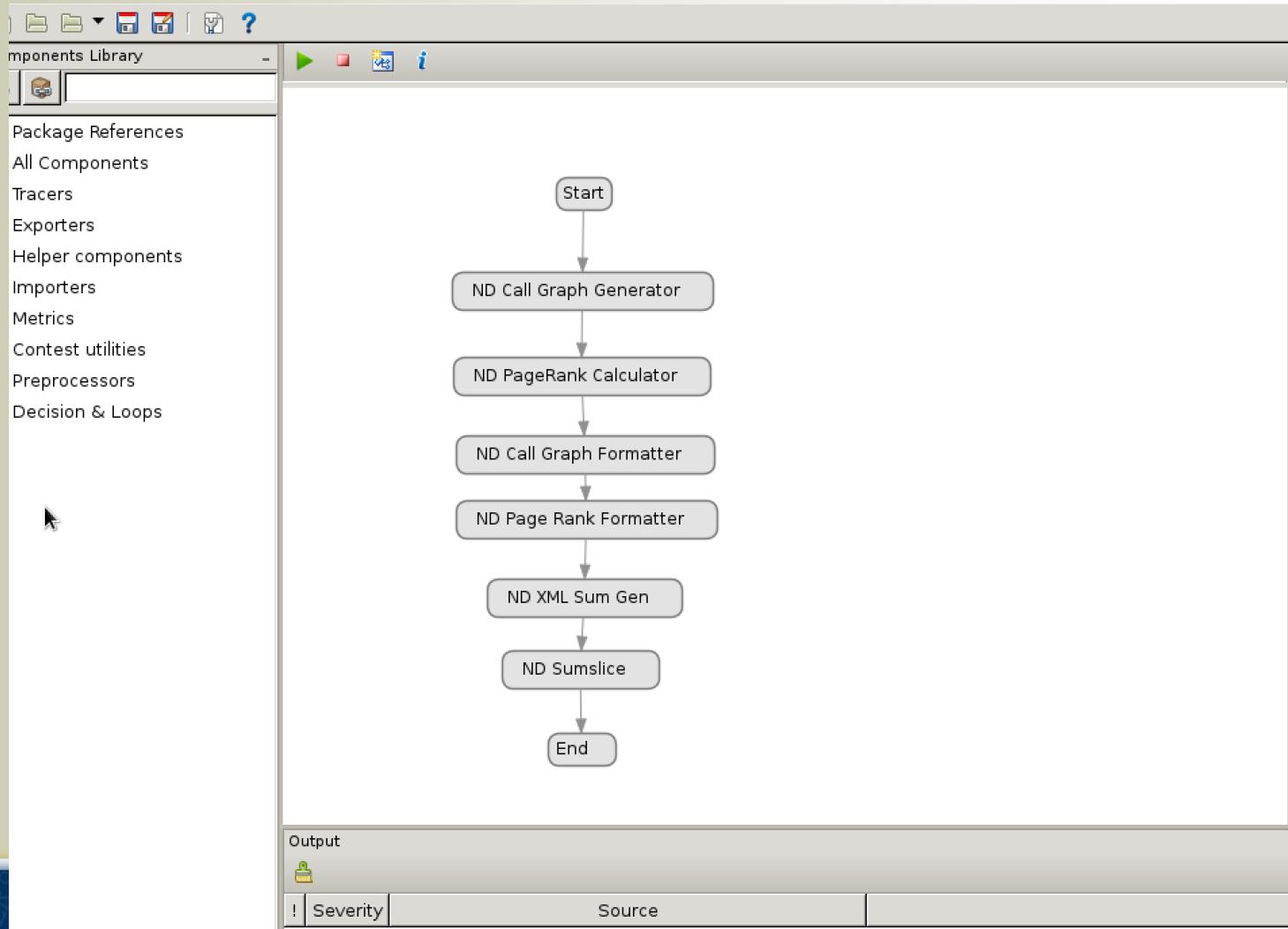
Abstractive Technique

- McBurney et al. Automatic Source Code Summarization of Context for Java Methods
- Provide high-level summaries of a method and its context
- Step-by-Step execution:
 - Collect contextual data about Java methods
 - Use key words to describe how a method is used
 - Use SWUM to identify parts-of-speech for the different keywords
 - Use PageRank to compute the call graph

Abstractive Technique

- McBurney et al. Automatic Source Code Summarization of Context for Java Methods
- Provide high-level summaries of a method and its context
- Step-by-Step execution:
 - Collect contextual data about Java methods
 - Use key words to describe how a method is used
 - Use SWUM to identify parts-of-speech for the different keywords
 - Use PageRank to compute the call graph
 - Generate a readable description of the context for each method in a java program

Natural Language Generation System



Input

```
<comment type="block" format="javadoc">/**  
 * This reader reads data from another reader until the end of a CDATA section  
 * (]]&gt;) has been encountered.  
 *  
 * @author Marc De Scheemaeker  
 * @version $Name: RELEASE_2_2_1 $, $Revision: 1.3 $  
 */</comment>  
<class> class <name>CDATAREader</name>  
  <super><extends>extends <name>Reader</name></extends></super>  
<block> {  
  
  <comment type="block" format="javadoc">/**  
 * The encapsulated reader.  
 */</comment>  
<decl_stmt><decl><specifier>private</specifier> <type><name>IXMLReader</name></type> <name>reader</name></decl>;</decl_stmt>  
  
<comment type="block" format="javadoc">/**  
 * Saved char.  
 */</comment>  
<decl_stmt><decl><specifier>private</specifier> <type><name>char</name></type> <name>savedChar</name></decl>;</decl_stmt>  
  
<comment type="block" format="javadoc">/**  
 * True if the end of the stream has been reached.  
 */</comment>  
<decl_stmt><decl><specifier>private</specifier> <type><name>boolean</name></type> <name>atEndOfData</name></decl>;</decl_stmt>  
  
<comment type="block" format="javadoc">/**  
 * Creates the reader.  
 *  
 * @param reader the encapsulated reader  
 */</comment>  
<constructor><name>CDATAREader</name><parameter_list>(<parameter><decl><type><name>IXMLReader</name></type> <name>reader</name></decl></parameter></parameter_list></constructor>  
<block> {
```

Output

-> QuickSummaryMessage-> ReturnMessage-> OutputUsedMessage-> OutputUsedMessage-> OutputUsedMessage-> OutputUsedMessage
This method nulls the null and returns a char. ErrorInvalidEntity() seems less important than average because it is not called by any methods.

This method creates the CDATA reader and returns a cdatareader. `CDataReader()` seems less important than average because it is not called by any meth

This method finalizes the CDATA reader. Finalize() seems less important than average because it is not called by any methods.

This method creates the XML parse exception and returns a `xmlparseexception`. `XMLParseException()` seems less important than average because it is not

This method reads the char and returns an int. Read() seems less important than average because it is not called by any methods.

This method closes the CDATA reader. Close() seems less important than average because it is not called by any methods.

This method creates the element and returns an IXMLElement. CreateElement() seems less important than average because it is not called by any method

This method creates the element and returns an IXMLElement. CreateElement() seems less important than average because it is not called by any method

This method creates the PC data element and returns an IXMLElement. CreatePCDataElement() seems less important than average because it is not called

This method gets the full name and returns a String. GetFullName() seems less important than average because it is not called by any methods.

This method gets the parent and returns an IXMLElement. GetParent() seems less important than average because it is not called by any methods.

This method creates the element and returns an IXMLElement. CreateElement() seems less important than average because it is not called by any method

This method creates the element and returns an IXMLElement. CreateElement() seems less important than average because it is not called by any method

This method sets the name. `SetName()` seems less important than average because it is not called by any methods.

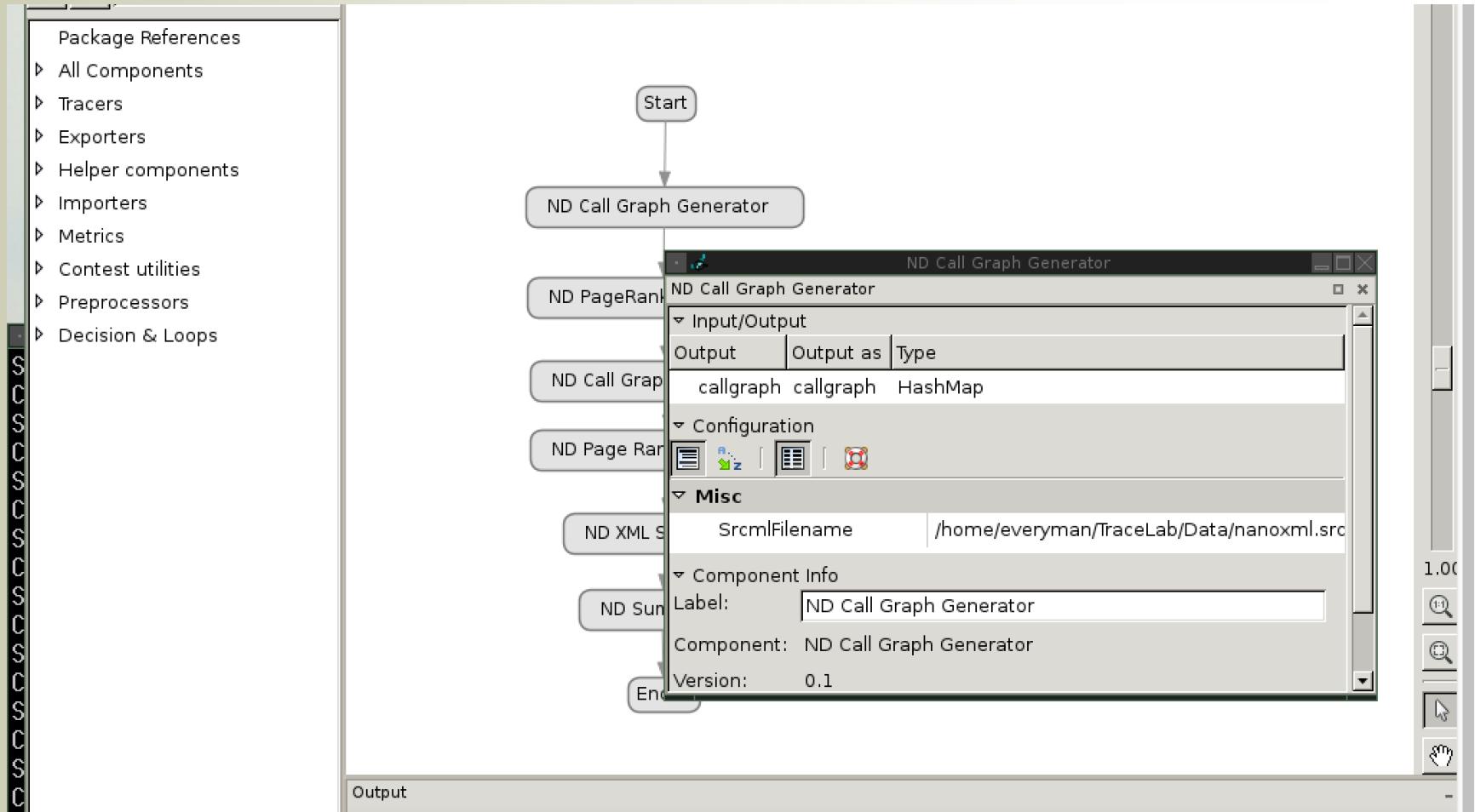
This method sets the name. SetName() seems less important than average because it is not called by any methods.

This method gets the namespace and returns a String. GetNamespace() seems less important than average because it is not called by any methods.

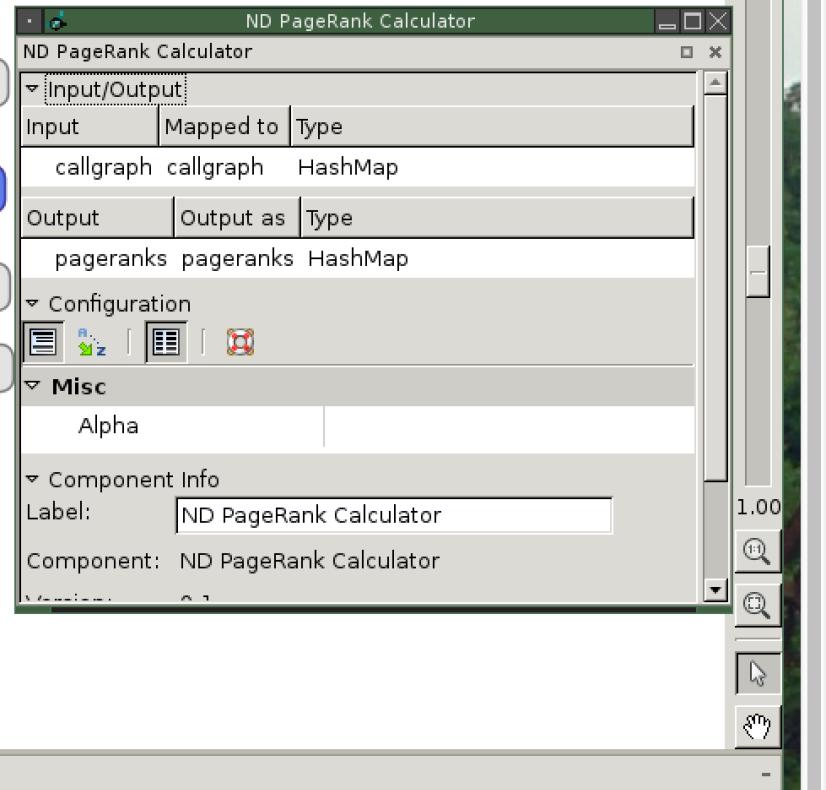
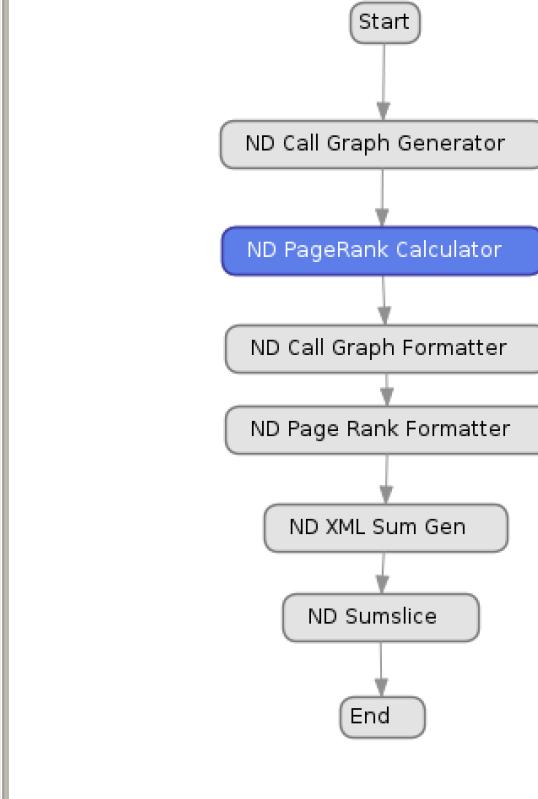
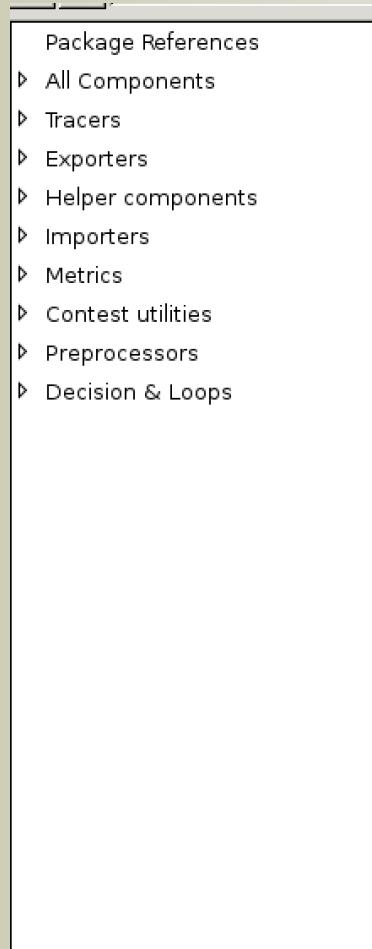
This method gets the name and returns a String. GetName() seems less important than average because it is not called by any methods.

This method enumerates the children and returns an Enumeration. `EnumerateChildren()` seems less important than average because it is not called by any

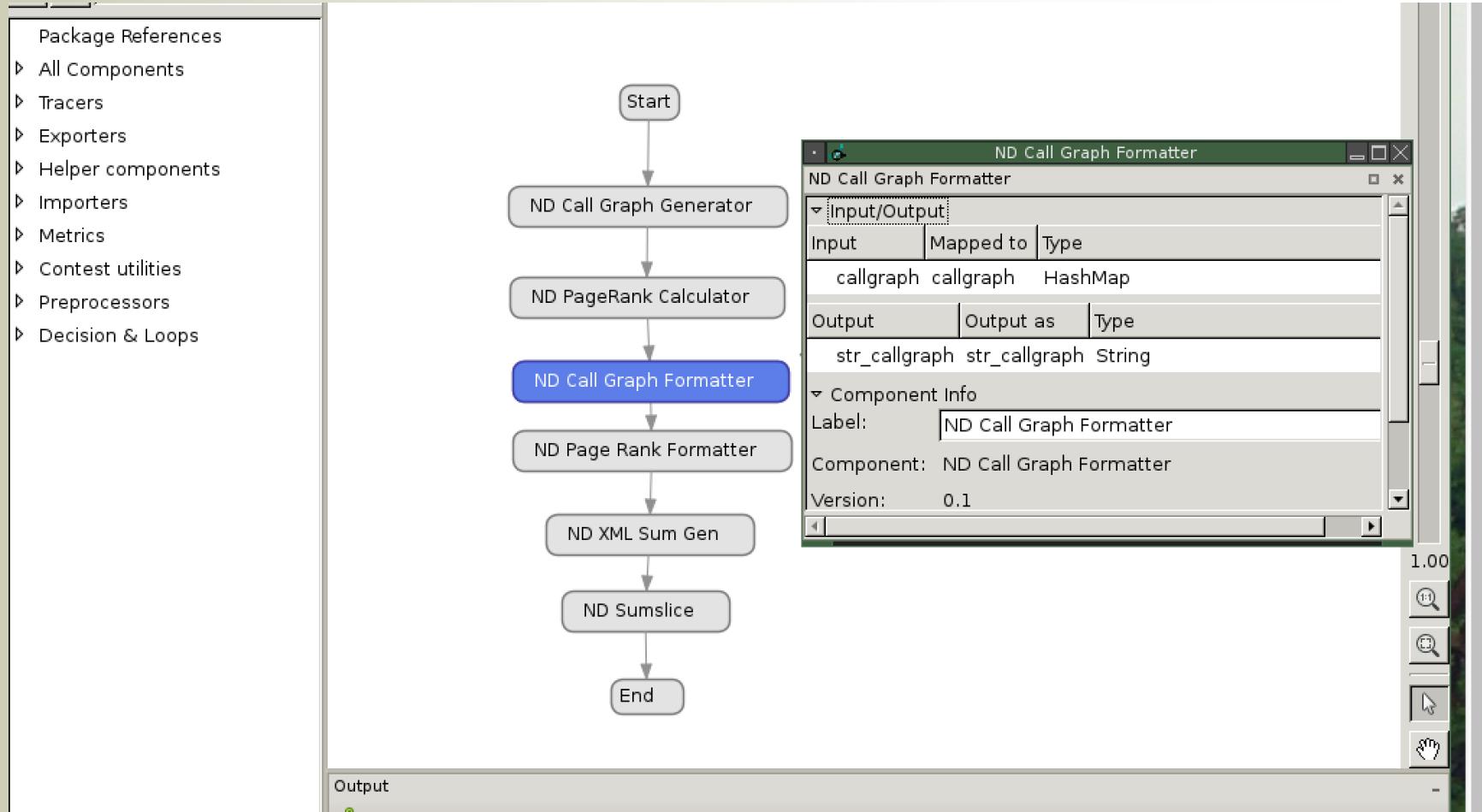
ND Call Graph Generator



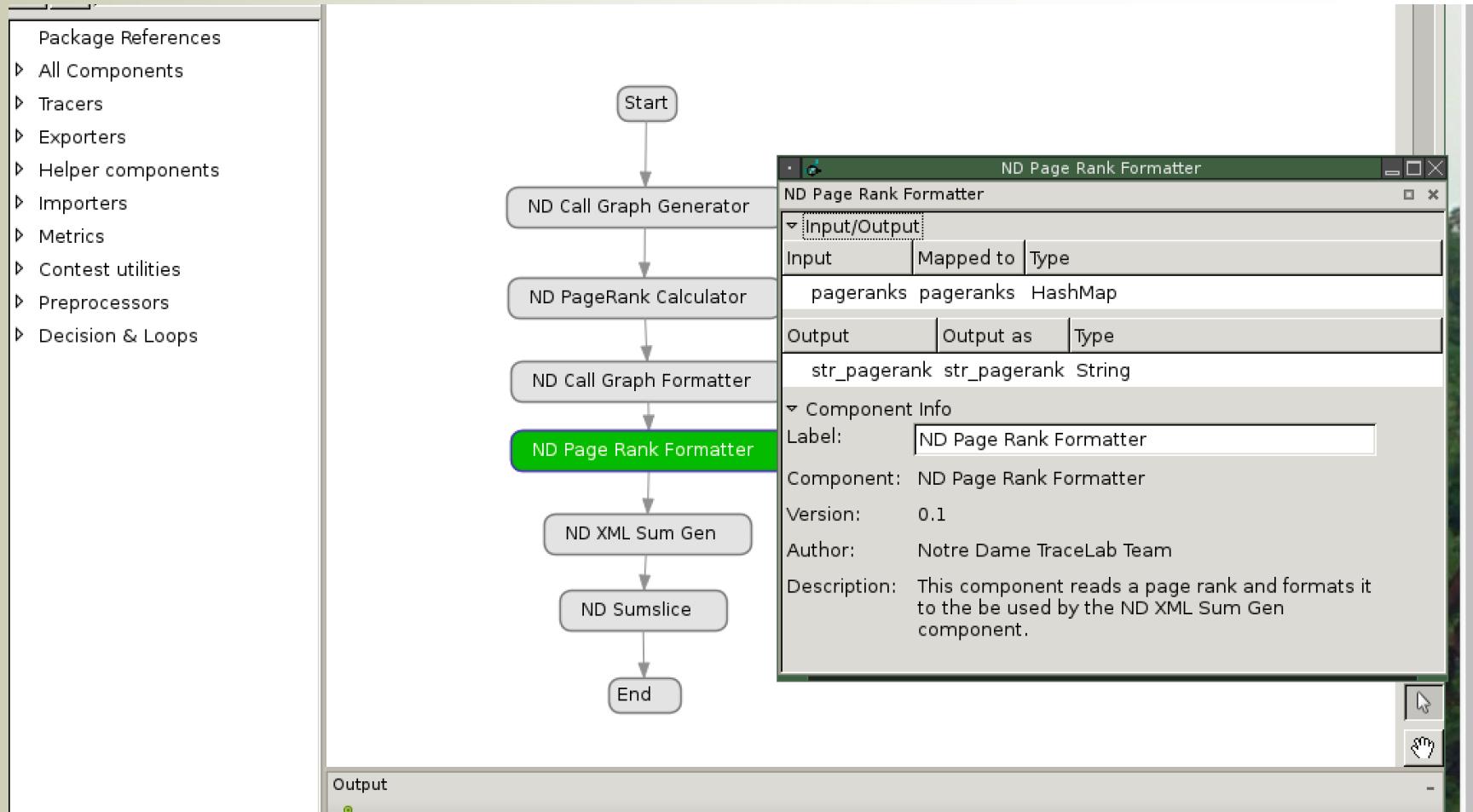
ND PageRank Calculator



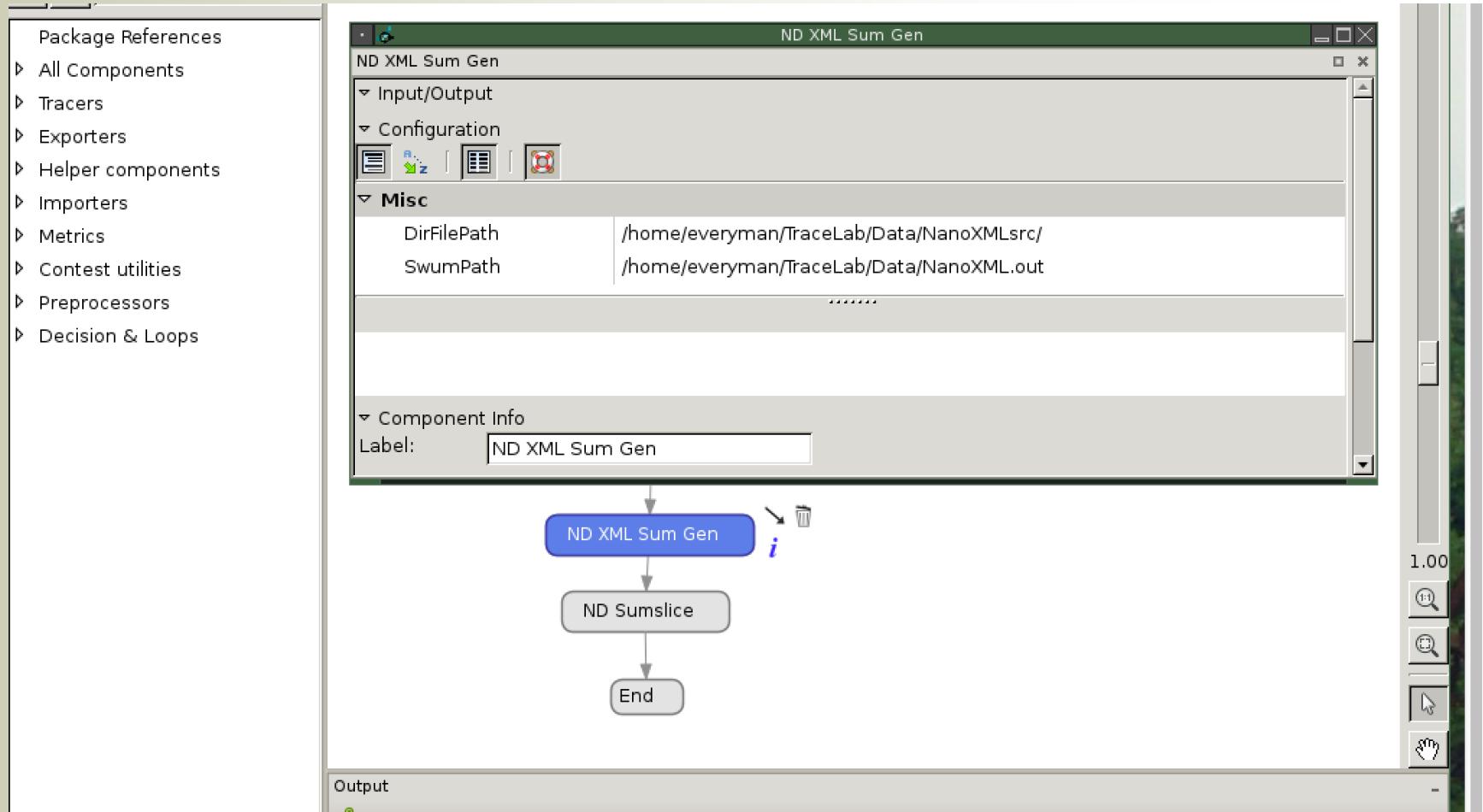
ND Call Graph Formatter



ND Page Rank Formatter



ND XML Sum Gen



ND XML Sum Gen Input

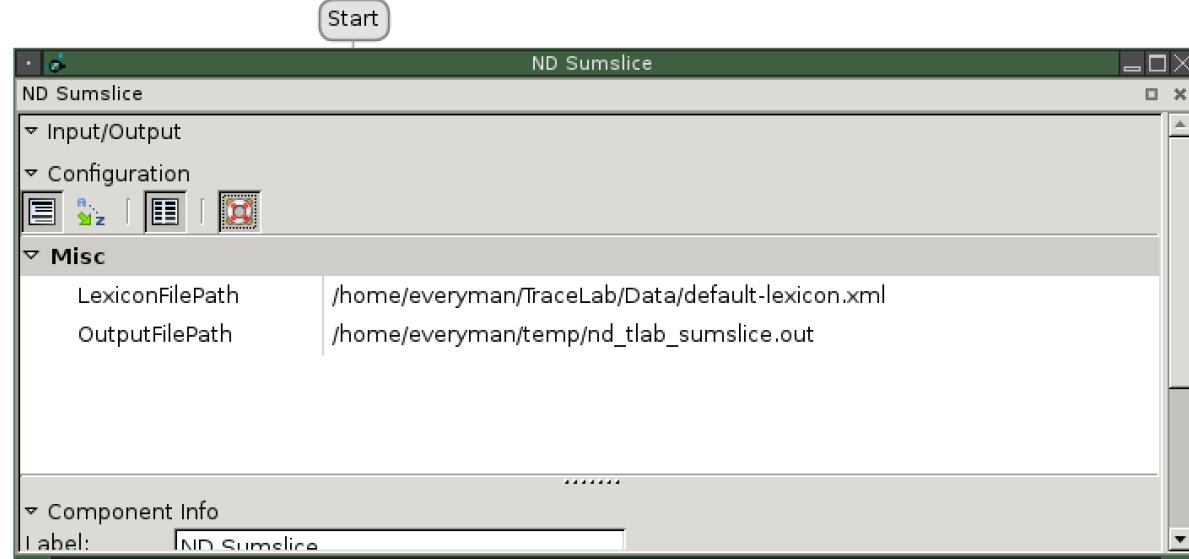
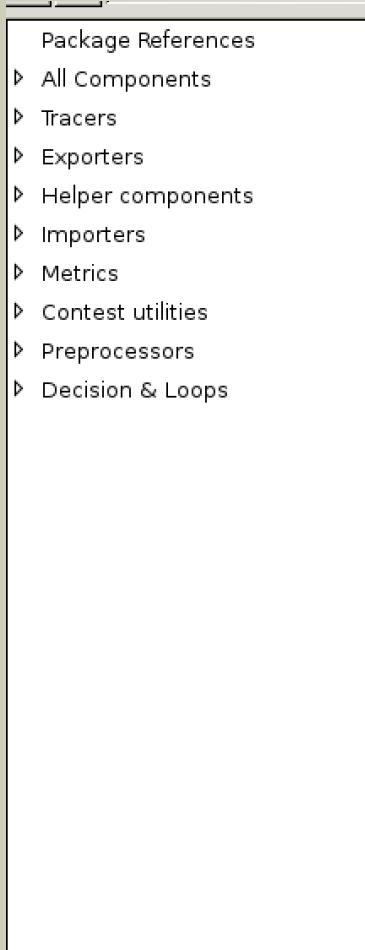
NanoXML.out - /home/everyman/TraceLab/Data

File Edit Search Preferences View Window Help

SPECIAL::main (V) || string (NI) - args (N)] ++ :: dump (NM) XML (N) ::DumpXML_static_void_main_String[]
SPECIAL::start (V) | building (N) ++ :: [string (NI) - system (NM) ID (NI)] :: [int (N) - line (NM) nr (N)] :: my (PR) builder (N) ::MyBuilder_void_startBuildin
VOID_NP::handle (V) | new (NM) processing (NM) instruction (N) ++ :: [string (NI) - target (N)] :: [reader (N) - reader (N)] :: my (PR) builder (N) :: void (N
SPECIAL::start (V) | element (NI) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: [string
SPECIAL::end (V) | element (NI) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: my (PF
SPECIAL::add (V) | attribute (N) ++ :: [string (NI) - key (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: [string (I
ENDS_WITH_ED::handle (V) | element (NM) attributes (NM) processed (N) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI)
SPECIAL::add (V) | PC (NM) data (NI) ++ :: [reader (N) - reader (N)] :: [string (NI) - system (NM) ID (NI)] :: [int (N) - line (NM) nr (N)] :: my (PR) builder
SPECIAL::get (V) | result (N) ++ :: object (NI) :: my (PR) builder (N) ::MyBuilder_Object_getResult
ENDS_WITH_ED::handle (V) | attribute (NM) added (N) ++ :: [string (NI) - key (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM)
SPECIAL::main (V) | [string (NI) - args (N)] ++ :: demo (N) ::Demo_static_void_main_String[]
SPECIAL::main (V) | [string (NI) - args (N)] ++ :: demo (N) ::Demo_static_void_main_String[]
SPECIAL::get (V) | entity (N) ++ :: [IXML (NM) reader (N) - reader (N)] :: [string (NI) - name (NI)] :: reader (N) :: my (PR) resolver (N) ::MyResolver_Readi
SPECIAL::get (V) | entity (N) ++ :: [IXML (NM) reader (N) - reader (N)] :: [string (NI) - name (NI)] :: reader (N) :: my (PR) resolver (N) ::MyResolver_Readi
SPECIAL::main (V) | [string (NI) - args (N)] ++ :: demo (N) ::Demo_static_void_main_String[]
CONSTRUCTOR: | document (NM) element (NI) ++ ::DocumentElement_DocumentElement_DocumentElement
SPECIAL::set (V) | attribute (N) ++ :: [string (NI) - attr (NM) name (NI)] :: [string (NI) - value (NI)] :: document (NM) element (NI) ::DocumentElement_
SPECIAL::add (V) | child (N) ++ :: [document (NM) element (NI) - elt (NI)] :: document (NM) element (NI) ::DocumentElement_void_addChild_Documen
FIELD:[properties (N) - attrs (N)]::DocumentElement_Properties_attrs
FIELD:[vector (N) - children (N)]::DocumentElement_Vector_children
SPECIAL::get (V) | title (N) ++ :: string (NI) :: chapter (N) ::Chapter_String_getTitle
SPECIAL::get (V) | ID (NI) ++ :: string (NI) :: chapter (N) ::Chapter_String_getID
SPECIAL::get (V) | paragraphs (N) ++ :: enumeration (N) :: chapter (N) ::Chapter Enumeration_getParagraphs
SPECIAL::main (V) | [string (NI) - param (N)] ++ :: XML (NM) 2 (NM) HTML (N) ::XML2HTML_static_void_main_String[]
SPECIAL::get (V) | content (N) ++ :: string (NI) :: paragraph (N) ::Paragraph_String_getContent
SPECIAL::get (V) | alignment (N) ++ :: int (N) :: paragraph (N) ::Paragraph_int_getAlignment
FIELD:[int (N) - LEFT (N)]::Paragraph_static_final_int_LEFT
FIELD:[int (N) - CENTER (N)]::Paragraph_static_final_int_CENTER
FIELD:[int (N) - RIGHT (N)]::Paragraph_static_final_int_RIGHT
FIELD:[hashtable (N) - alignments (N)]::Paragraph_static_final_HashTable_alignments
SPECIAL::start (V) | building (N) ++ :: [string (NI) - system (NM) ID (NI)] :: [int (N) - line (NM) nr (N)] :: document (NM) builder (N) ::DocumentBuilder_v
VOID_NP::handle (V) | new (NM) processing (NM) instruction (N) ++ :: [string (NI) - target (N)] :: [reader (N) - reader (N)] :: document (NM) builder (N) ::
SPECIAL::start (V) | element (NI) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: [string
SPECIAL::end (V) | element (NI) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: docum
SPECIAL::add (V) | attribute (N) ++ :: [string (NI) - key (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI) - ns (NM) system (NM) ID (NI)] :: [string (I
ENDS_WITH_ED::handle (V) | element (NM) attributes (NM) processed (N) ++ :: [string (NI) - name (NI)] :: [string (NI) - ns (NM) prefix (N)] :: [string (NI)
SPECIAL::add (V) | PC (NM) data (NI) ++ :: [reader (N) - reader (N)] :: [string (NI) - system (NM) ID (NI)] :: [int (N) - line (NM) nr (N)] :: document (NM)
SPECIAL::get (V) | result (N) ++ :: object (NI) :: document (NM) builder (N) ::DocumentBuilder_Object_getResult
FIELD:[hashtable (N) - classes (N)]::DocumentBuilder_static_HashTable_classes
FIELD:[stack (N) - elements (N)]::DocumentBuilder_Stack_elements
FIELD:[document (NM) element (NI) - top (NM) element (NI)]::DocumentBuilder_DocumentElement_topElement
CONSTRUCTOR: | CDATA (NM) reader (N) ++ :: [IXML (NM) reader (N) - reader (N)]::CDATAReader_CDATAReader_CDATAReader__IXMLReader__
BASE_VERB::finalize (V) | CDATA (NM) reader (N) ++ ::CDATAReader_void_finalize

one everyman@debia (*) full : /home/ev ND Sumslice everyman@debia ND XML Sum Gen NanoXML.out - October 6 1:41 PM

ND Sumslice



ND Sumslice Input

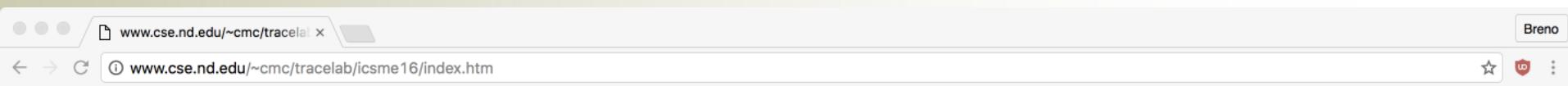


A screenshot of a desktop environment showing a window titled "default-lexicon.xml - /home/everyman/TraceLab/Data". The window contains XML code representing a lexicon. The XML structure includes elements like <lexicon>, <word>, <base>, <category>, <id>, <ditransitive/>, <transitive/>, <ability>, <adjective>, <predicative/>, <qualitative/>, and <abnormal>. The code is organized into multiple <word> blocks, each containing a <base> element and various category and ID attributes. The XML uses '>^M' as a line separator.

```
<lexicon>^M
<word>^M
<base>a</base>^M
<category>determiner</category>^M
<id>E0006419</id>^M
</word>^M
<word>^M
<base>abandon</base>^M
<category>verb</category>^M
<id>E0006429</id>^M
<ditransitive/>^M
<transitive/>^M
</word>^M
<word>^M
<base>abbey</base>^M
<category>noun</category>^M
<id>E0203496</id>^M
</word>^M
<word>^M
<base>ability</base>^M
<category>noun</category>^M
<id>E0006490</id>^M
</word>^M
<word>^M
<base>able</base>^M
<category>adjective</category>^M
<id>E0006510</id>^M
<predicative/>^M
<qualitative/>^M
</word>^M
<word>^M
<base>abnormal</base>^M
<category>adjective</category>^M
<id>E0006517</id>^M
<predicative/>^M
<qualitative/>^M
</word>^M
<word>^M
<base>abolish</base>^M
<category>verb</category>^M
<id>E0006524</id>^M
<transitive/>^M
</word>^M
<word>^M
```

Web Page

- <http://www.cse.nd.edu/~cmc/tracelab/icsme16/index.htm>



Resources

Please use the resources on this page to better understand and use our approach. All files can be downloaded [here](#). If you need additional help or information, feel free to contact [Dr. Collin McMillan](#).

We provided two virtual machines that contains the components, TraceLab, and a sample experiment. The first virtual machine (ubu) contains a working TFIDF experiment and the second virtual machine (TraceLab Debian) contains both working experiments. Fell free to download it and test our approach.

Virtual Machine Information:

VM1: ubu

- Size: 14.9GB
- OS: Ubuntu 12.04 (64bits)
- Memory: 2048MB
- User: smiley
- Password: password
- Download:[link](#)

VM1: TraceLab Debian

- Size: 4.4GB
- OS: Debian (64bits)
- Memory: 2048MB
- User: everyman
- Password: password
- Download: [link](#)
- Note: Log in via the command line, then type "startx" to start the GUI. Once in the GUI, right click and select "xterm". Then type "tracelab" to start TraceLab. The walkthroughs assume you have TraceLab open.

How to Install for Experts

1. Download the desired component
2. Update the following lines in the build.xml file: 'Components output directory', 'Path to IKVM directory', 'Path to TraceLabSDK.dll' and 'Path to TraceLabSDK.Types.dll'.
3. Run 'ant'
4. The component should be the ready to use

How to Install for Newcomers

Click [here](#) for a step-by-step guide on how to install a component to TraceLab.