

# jInfer AutoEditor Module Description

Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, Matej Vitásek  
Advisors: RNDr. Irena Mlýnková, Ph.D., Martin Nečaský, Ph.D.

Praha, 2011

Target audience: developers willing to extend jInfer, specifically alter displaying of automata .

Responsible developer:	Mário Mikula
Required tokens:	org.openide.windows.WindowManager
Provided tokens:	none
Module dependencies:	Base JUNG
Public packages:	cz.cuni.mff.ksi.jinfer.autoeditor cz.cuni.mff.ksi.jinfer.autoeditor.automatonvisualizer cz.cuni.mff.ksi.jinfer.autoeditor.automatonvisualizer.layouts cz.cuni.mff.ksi.jinfer.autoeditor.gui.component

## 1 Introduction

This is an implementation of a *AutoEditor*. Using JUNG library, it provides an API to display and user interactively modify automata, so the process of inference can be easily made user interactive.

## 2 Structure

Structure of *AutoEditor* can be divided into following four main parts.

- API - API to display automaton in GUI.
- Base classes - Classes providing basic functionality that can be extended and combined to achieve desired visualization of an automaton.
- Derived classes - Classes derived from the base classes that are used in existing modules and simultaneously serve as examples.
- Layout factory - TODO

First, base classes and a creation of automaton visualization will be described.

### 2.1 Base classes

Main two classes representing visualization of automaton are *Visualizer* and *AbstractComponent*. *Visualizer* is a graphical representation of automaton and *AbstractComponent* is a panel (extends *JPanel*) containing the *Visualizer* which will be displayed in GUI. TODO obrazok ako AC dedi od *JPanelu* a obsahuje *Visualizer*.

#### 2.1.1 Visualizer

TODO translate

Trieda *Visualizer* dedi od *JUNG*oveho *VisualizationViewer*, takže poskytuje všetky jeho metódy a navyše podporu pre uloženie automatu do obrázku - metódy *saveImage()* a *getSupportedImageFormatNames()*. Pre uloženie obrázku však tieto metódy nie je nutné volať, pretože *AutoEditor* GUI obsahuje tlačidlo na uloženie práve vykresleného automatu do obrázku (viz ďalej).

Constructor method bere ako argument instanciu triedy Layout. Viac informacii o tejto triede a jej pouzitie v kapitole TODO ref.

TODO obrazok ako Visualizer dedi od VisualizationVieweru a obsahuje Layout.

### 2.1.2 PluggableVisualizer

Tried PluggableVisualizer je rozsiazenim triedy Visualizer, ktora poskytuje navyse jednoduchy sposob ako zapajat mouse pluginy.

Mouse plugin je trieda JUNGu, ktore rozsiruje moznosti Visualizera o pracu s mysou (vyberanie stavov, zoom, ...). Spomenut uz existujuce pluginy a ref.

TODO obrazok ako PluggableVisualizer dedi od Visualizeru

Metody:

addGraphMousePlugin() setVertexLabelTransformer() replaceVertexLabelTransformer() setEdgeLabelTransformer() replaceEdgeLabelTransformer()

By default obsahuje 2 pluginy, jeden pre zoom a jeden pre posuvanie canvasu. V pripade potreby je pozne ich odstranit pomocou metod VisualizationVieweru. TODO dopisat.

### 2.1.3 AbstractComponent

TODO translate

Trieda AbstractComponent je panel v ktorom bude vykresleny automat, presnejsie Visualizer reprezentujuci nejaký automat. Dedi od triedy JPanel, takže poskytuje všetky jej metódy a správanie. Navyše poskytuje metódy

setVisualizer() getVisualizer() waitForGuiDone() guiDone() guiInterrupt() guiInterrupted()

a abstrakt metódu getAutomatonDrawPanel()

Purpose tejto triedy je rozšíriť ju a poskladať si panel aký sa hodi (tlacitka, napisy, ...) s tým, že musí obsahovať aspoň jeden JPanel, v ktorom bude vykreslený nastavený Visualizer. Účel metódy getAutomatonDrawPanel() je vrátiť tento JPanel, aby AutoEditor vedel, kam má ten Visualizer vykresliť.

Ak je žiadaný user interaktivita, je nutné si podporu pre ňu zahrnúť práve do tejto triedy. Pre viac informácií viz TODO ref.

Visualizer sa nenastavuje v konštruktor z toho dôvodu, že často je žiaduce, aby sa na rovnakom paneli kreslilo postupne viac rôznych automatov. Na to nie je nutné vyrábať novú instanciu, ale stačí na jednej instancii volať setVisualizer().

## 2.2 API

API AE je veľmi jednoduché. Trieda AutoEditor poskytuje tieto 3 statické metódy.

drawComponentAsync() drawComponentAndWaitForGUI() closeTab()

## 2.3 Derived classes

Popis tried použitých v iných moduloch, ktoré slúžia zároveň ako príklad.

StatePickingVisualizer StatesPickingVisualizer

## 2.4 Layout factory

TODO

## 2.5 GUI

TODO

tlacitka

## 2.6 Preferences

TODO

All settings provided by *BasicXSDExporter* are project-wide, the preferences panel is in `cz.cuni.mff.ksi.jinfer.basicxsd.properties` package. As mentioned above, it is possible to set the following.

- Turn off generation of global element types. Turning off this feature is not recommended as it may cause certain problems with validity of resulting XSD. See ??.
- Minimal number of occurrences of element to define its type globally. (Only if generation of global elements is active.)
- Number of spaces in output per one level of indentation.
- Global type name prefix. It is a string which will be inserted before a name of a type, which is derived from element's name. Can be also an empty string. (Only if generation of global elements is active.)
- Global type name suffix. It is a string which will be appended after a name of a type, which is derived from element's name. Can be also an empty string. (Only if generation of global elements is active.)

## References

- [Aho96] H. Ahonen. *Generating grammars for structured documents using grammatical inference methods*. PhD thesis, Department of Computer Science, University of Helsinki, Series of Publications A, Report A-1996-4, 1996.
- [Bou] Ronald Bourret. Dtd parser, version 2.0. <http://www.rpbourret.com/dtdparser/index.htm>.
- [HMU01] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (2nd Edition)*. Addison-Wesley, 2001.
- [HW07] Yo-Sub Han and Derick Wood. Obtaining shorter regular expressions from finite-state automata. *Theor. Comput. Sci.*, 370(1-3):110–120, 2007.
- [jun] Java universal network/graph framework. <http://jung.sourceforge.net/>.
- [KMS<sup>+</sup>a] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer Architecture*.
- [KMS<sup>+</sup>b] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer AutoEditor automaton visualization and editor module*.
- [KMS<sup>+</sup>c] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer Base Module Description*.
- [KMS<sup>+</sup>d] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicDTDExporter Module Description*.
- [KMS<sup>+</sup>e] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicIGG Module Description*.
- [KMS<sup>+</sup>f] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicRuleDisplayer Module Description*.
- [KMS<sup>+</sup>g] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jinfer javadoc*. <http://jinfer.sourceforge.net/javadoc>.
- [KMS<sup>+</sup>h] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer TwoStep simplifier design and implementation*.
- [log] Apache log4j<sup>TM</sup>. <http://logging.apache.org/log4j/>.
- [loo] org.openide.util.class lookup. <http://bits.netbeans.org/dev/javadoc/org-openide-modules/org-openide/modules/doc-files/api.html>.
- [mod] Module system api. <http://bits.netbeans.org/dev/javadoc/org-openide-modules/org-openide/modules/doc-files/api.html>.
- [Nor] Theodore Norvell. A short introduction to regular expressions and context free grammars. <http://www.engr.mun.ca/~theo/Courses/fm/pub/context-free.pdf>.
- [VMP08] Ondřej Vošta, Irena Mlýnková, and Jaroslav Pokorný. Even an ant can create an xsd. In *DASFAA'08: Proceedings of the 13th international conference on Database systems for advanced applications*, pages 35–50, Berlin, Heidelberg, 2008. Springer-Verlag.
- [wik] Regular expression. [http://en.wikipedia.org/wiki/Regular\\_expression](http://en.wikipedia.org/wiki/Regular_expression).