jInfer BasicRuleDisplayer 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, looking for ways to visualize grammars.

Responsible developer:	Matej Vitásek
Required tokens:	none
Provided tokens:	cz.cuni.mff.ksi.jinfer.base.interfaces.RuleDisplayer
Module dependencies:	Base
Public packages:	none

1 Introduction

This is fairly basic rule displayer draws rules as series of nested coloured rectangles. It does so by recursively descending down the regexp tree to draw the innermost nodes first, composing the outer nodes from them. User interface-wise, it creates a component with multiple panels: each one for a new grammar to display.

tabs

2 Structure

The main class implementing `RuleDisplayer` inference interface and simultaneously registered as its service provider is `BasicRuleDisplayer`. Main method of this class is `createDisplayer()`, which looks up the component, adds a new panel to it and renders the specified grammar in it.

The diagram of classes involved in rule display is in figure 1. The top component of the rule displayer has a few panels in it: each contains its own `RuleDisplayer` responsible for rendering specified grammar once into an internal `Image`, then keeping it and drawing it onto panel's `Canvas` each time repainting is needed.

Grammar is rendered in the `drawRules()` method by recursively rendering its rules and stacking their image representations into the image. Rule painting is handled in the `NodePainter` class, with most important method `drawNode()`. While the code itself is a nice recursion programming exercise, there is nothing of a special interest in it.

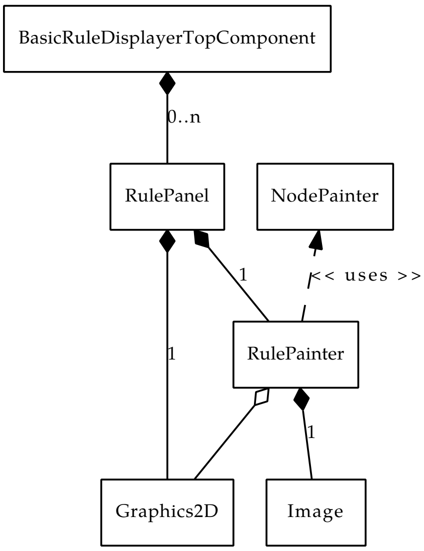
All graphics used by *BasicRuleDisplayer* is contained in the `cz.cuni.mff.ksi.jinfer.basicruledisplayer.graphics` package.

2.1 Settings

All settings provided by *BasicRuleDisplayer* are NetBeans-wide. The options panel along with all the logic is in the `cz.cuni.mff.ksi.jinfer.basicruledisplayer.options` package. Available options include setting the maximum number of panels concurrently displayed, rules drawn, as well as maximum nesting level. Everything beyond these limits will be rendered as an icon suggesting omitted information. Furthermore, colours of various nodes can be set here.

~~[Aho96]~~

i've solved this already



dia uses dashed as 'implements' blueash

why diamonds are not all empty?

Figure 1: BasicRuleDisplayer class diagram

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.