jInfer ProjectType 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 extend jInfer project structure.

Responsible developer:	Michal Švirec
Required tokens:	cz.cuni.mff.ksi.jinfer.base.interfaces.inference.IGGenerator cz.cuni.mff.ksi.jinfer.base.interfaces.inference.SchemaGenerator cz.cuni.mff.ksi.jinfer.base.interfaces.inference.Simplifier org.openide.windows.IOProvider
Provided tokens:	none
Module dependencies:	Base Runner
Public packages:	cz.cuni.mff.ksi.jinfer.projecttype.actions

1 Introduction

`ProjectType` is the module responsible for creation of NBP project type which groups input/output files that belongs to one specific inference. Each jInfer project also allows user to set properties specific for inference.

With each project created in jInfer is also created specific directory structure on filesystem that describes jInfer project. This structure is described in figure 1, where rectangles represent folders and rounded rectangles files. Folder in filesystem is considered as jInfer project folder, if contains *jinferproject* subfolder. These folder contains two files, first is *project.properties*, where all properties set for particular project are saved. Second file is *input.files* which is xml file filled with paths to files inserted as input into particular project (structure of this file is described in section 2.1.3).

2 Structure

Structure of *`ProjectType`* can be divided into following five main parts.

- Base classes - Classes providing main functionality like creation of project, defining operations like move, delete, copy etc. All the base classes are contained in the `cz.cuni.mff.ksi.jinfer.projecttype` package.

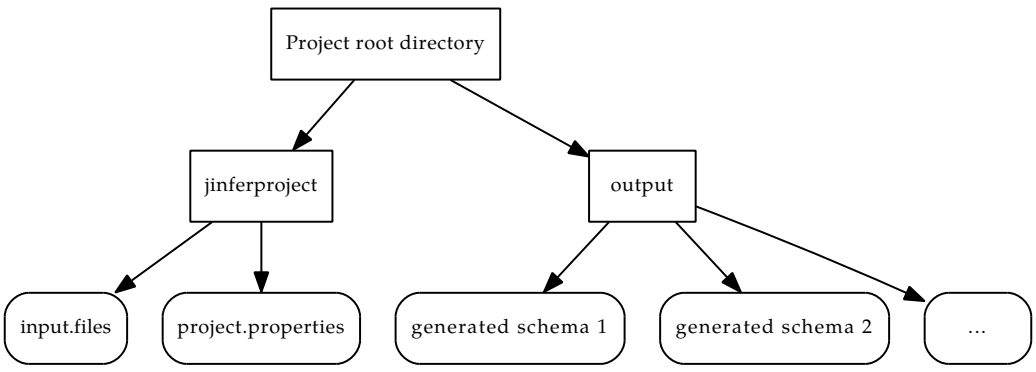

Figure 1: Project directory structure

- Visualization classes - These classes create tree structure of project in NBP Projects view and are contained in the `cz.cuni.mff.ksi.jinfer.projecttype.nodes`
- Actions - Classes from `cz.cuni.mff.ksi.jinfer.projecttype.actions` package that provides actions allowing adding, removing input files into project, or running the project.
- Properties - Classes responsible for creation of project properties window with properties category tree. These are situated in `cz.cuni.mff.ksi.jinfer.projecttype.properties` package.
- Project wizard - Classes representing creation of project through new project wizard from `cz.cuni.mff.ksi.jinfer.projecttype.sample` package.

2.1 Base classes

This section describes classes needed for creation of the jInfer project and correct integration into NBP. Main two classes representing jInfer project type are `JInferProjectFactory` and `JInferProject`.

2.1.1 JInferProjectFactory

`JInferProjectFactory` is a factory class implementing `ProjectFactory` interface provided by NBP. This class is responsible for loading/saving of jInfer project from/to filesystem and also determining if folder in filesystem is type of jInfer project (contains *jinferproject* subfolder). For this purpose, class has three methods: `isProject()`, `save()` and `load()`. When jInfer project is loaded from filesystem, `JInferProjectFactory` creates new instance of `JInferProject` passing path to project folder as a constructor parameter. When `save()` method is invoced, all the project properties are saved in *projectproperties* file and paths to input files are saved in *input.files*.

2.1.2 JInferProject

`JinferProject` class implements `Project` interface from NBP and represents in-memory representation of jInfer project. Inner structure of this class is very simple, interface provides only two methods: `getProjectDirectory()` and `getLookup()`. All the extensibility of project type is done by *Lookup* functionality of NBP. Project lookup contains besides properties and `Input` class, which encapsulates input files, also class that creates output files, class that builds project tree in project window, etc.

2.1.3 InputFiles class

`InputFiles` is utility class that stores/loads input file paths into/from input.files xml file. For this purpose, class uses *JAXB* architecture to unmarshall/marshall xml file into/from java content objects. Xml structure of the input.files is very simple, *jinferinput* is root element under which are *xml*, *schemas* and *queries* elements. Each of this elements could have none or more *file* elements with string attribute *loc* that contains absolute path to input file.

Example of the structure follows.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jinferinput>
    <xml>
        <file loc="/home/user/example.xml"/>
    </xml>
    <schemas>
<file loc="/home/user/example.dtd"/>
    </schemas>
    </queries/>
</jinferinput>
```

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`.

[gra] Graph visualization software. `http://www.graphviz.org/`.

[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+a] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer Architecture*.

[KMS+b] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer AutoEditor automaton visualization and editor module*.

[KMS+c] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer Base Module Description*.

[KMS+d] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicDTDExporter Module Description*.

[KMS+e] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicIGG Module Description*.

[KMS+f] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer BasicRuleDisplayer Module Description*.

[KMS+g] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. jinfer javadoc. `http://jinfer.sourceforge.net/javadoc`.

[KMS+h] Michal Klempa, Mário Mikula, Robert Smetana, Michal Švirec, and Matej Vitásek. *jInfer TwoStep simplifier design and implementation*.

[log] Apache log4j™. `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.

[Vyh] Julie Vyhňanovská. Automatic construction of an xml schema for a given set of xml documents.

[wik] Regular expression. `http://en.wikipedia.org/wiki/Regular_expression`.