PRESENTATION 14

Department of Computer Science
University of Petrozavodsk

The Web-SynDic System: Input Data Translation, Session Management and Activity Statistics

Andrey Y. Salo

Abstract

The Web-SynDic project is a student software engineering (SE) project of the Petrozavodsk State University (PetrSU), Department of Computer Science (CSDept). The project is also held in the framework of cooperation between the CS Departments of PetrSU and the University of Helsinki (UH). CSDept of UH helps the PetrSU student team to study SE standards and technology and makes an expert estimation of the process.

The project is related to the research done at CSDept of PetrSU. The research deals with the development of a new type of algorithms for efficient solving some classes of nonnegative linear Diophantine equations (NLDE) by syntactic (parsing) methods. The objectives of the project include the following.

- A need of a web system to present current research results on the syntactic algorithms;
- A practical exercise for PetrSU students in software engineering standards and technology;
- Training the PetrSU students to participate in a joint distributed (via Internet) SE project with UH students (it is started at January 2004).

The waterfall model is chosen for the software process. Requirements analysis is performed by the whole team. During "Design", "Implementation", and "Testing" phases each developer has a number of assigned system modules. My part consists of the following modules:

1. Input data translators;
2. Session management subsystem;
3. Activity statistics subsystem.

Input Data Translators

One of the most important parts of the user interface is input data conversion. The basic input data consists of ANLDE systems and ANLDE system sets. The input data translators, which are used to convert ANLDE systems and ANLDE system sets from traditional mathematical style to internal memory data structures, are described in this section.

ANLDE System Format

Format:

```
# Comment
x1 + x2 + ...  + xK2 = c11*x1 + c12*x2 + ...  + c1N*xN
x[K2+1] + x[K2+2] + ...  + xK3 = c21*x1 + c22*x2 + ...  + c2N*xN
...
x[KM+1] + x[KM+2] + ...  + xN = cM1*x1 + cM2*x2 + ...  + cMN*xN
```

Description: The format represents an ANLDE system. c11, c12, ..., c1N, c21, c22, ..., cMN are coefficients (optional, default value is 1). x1, x2, ..., xN are unknowns, may appear in any order, some may be skipped. If there is no unknowns after the "=" sign, write "0". Each unknown must appear in the left-hand side of some equation at most one time. Blank and comment lines are ignored.

Sample:

```
# Sample ANLDE system
x1 + x4 = 2*x1 + 3*x3
x2 + x3 = x1 + 2*x2 + x3
```

ANLDE System Set Format

Format:

```
<ANLDE system 1>
%
<ANLDE system 2>
%
...
<ANLDE System N>
```

Description: The format represents ANLDE System Set ⟨ANLDE System 1⟩, ..., ⟨ANLDE System N⟩. Each system is in the ANLDE System Input Format (sect.). Blank and comment lines are ignored. String with symbols(s) '%' is a delimiter for ANLDE systems. These strings may additionally contain blank symbols ('␣', '\t') only.

ANLDE System Parser Generation

ANLDE system parser is constructed from two files containing specifications for syntax analyzer and lexical analyzer. Files with specifications are translated to Java source files using

Byacc/J and JFlex. The generated class ANLDESystemParser is used by class ANLDEParser to translate single ANLDE systems. The generated class ANLDESystemLexer is used by ANLDESystemParser as a scanner.

Flex specification:

```
INT = [0-9]+                    /* integer value */
VAR = [A-Za-z][A-Za-z0-9]*      /* variable name */
NL  = [\r\n]+                   /* newline */
```

YACC specification:

```
input:                      /* empty input */
    | input line
    ;

line: NL
    | left '=' right NL     /* a line containing an equation */
    | left '=' INT NL       /* a line containing an equation
                               with zero right-hand part */
    ;

left: VAR                   /* rule for left-hand part of ANLDE */
    | left '+' VAR
    ;

right: item                 /* rule for right-hand part of ANLDE */
    | right '+' item
    ;

item: VAR                   /* coefficient before variable equals to 1 */
    | INT '*' VAR           /* coefficient before variable equals to INT */
    ;
```

If an error has been matched while parsing, ANLDEFormatException containing error description and line number is thrown. If during the translation coefficients, equations, or unknowns limit has been exceeded, ANLDEFormatException is also thrown.

The generation process consists of two steps:

1. The file containing Flex specification of lexical scanner is passed to JFlex tool and ANLDESystemLexer class is generated.

2. The file containing Yacc specification of syntax analyzer which uses ANLDESystem-Lexer as lexical scanner are passed to Byacc/J tool and ANLDESystemParser class is generated.

Session Management

One of the main concept of the Web-SynDic system is session. There are two types of sessions: a physical server session and a logical Web-SynDic session. Web-SynDic session stores all information about current work in the system, for example ANLDE systems, user profile etc. Web-SynDic session is a continuous time period of user's working with the Web-SynDic. Further we will use term "session" instead of "Web-SynDic session".

Session is established when any user (regular on registered) start to use the system. It is open while a user logs out or closes the connection. If a user logs out, then the session is closed at once, but if a user closes the connection, for example closes the browser window, the session is still alive for a some period of time (default is 15 min and configured by Jakarta Tomcat server).

If a user has established a session and does not use the system with a large time period the session will be terminated automatically. For using the web-system a user should establish a session again.

Each session has own identifier on the server side. A new identifier of session is generated whenever the session has been established.

If cookies are enabled in user's browser, then the identifier of server session is passed by cookies. If they are disabled the session identifier is passed by the URL string in a browser window.

Class SessionManager is used to store user profile, limits, statistics and last solved ANLDE system information. It is shown in Fig. 1.

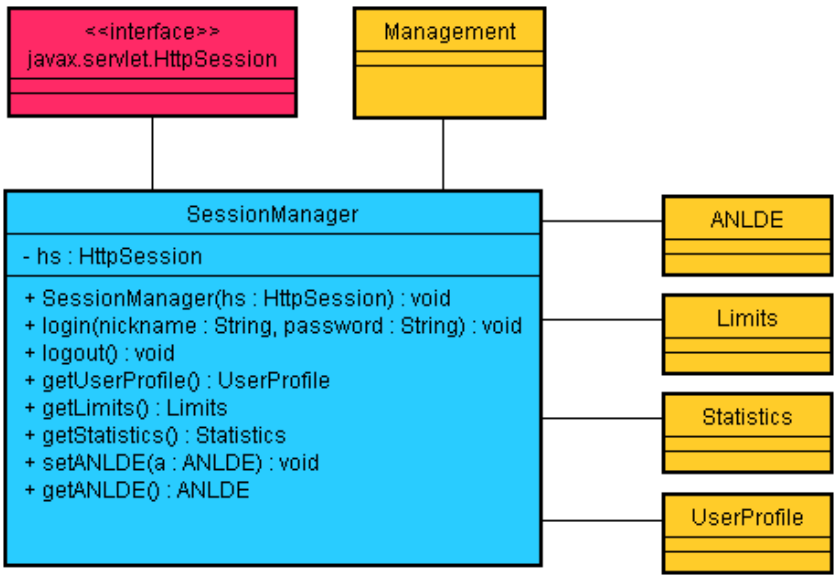

Figure 1: Class SessionManager

Activity Statistics

Administrator has ability to view activity statistics, collected by the web system. The activity statistics mean summarized information appropriate to selected domain and metrics for current month. Activity statistics domain is a criterion by which statistics information is summarized. Nickname or IP address can be selected as the domain of the report. Activity statistics metrics define categories for statistics report. Number of sessions, total sessions time, requests for solving, requests for generation, total system time, total work time, agreements with solutions can be selected as metrics for a required report. A statistics report contains time of report generation and statistics information appropriate to selected domain and metrics for current month. Records in the report are sorted by domain.

The class diagram used in activity statistics subsystem is shown in Fig. 2.

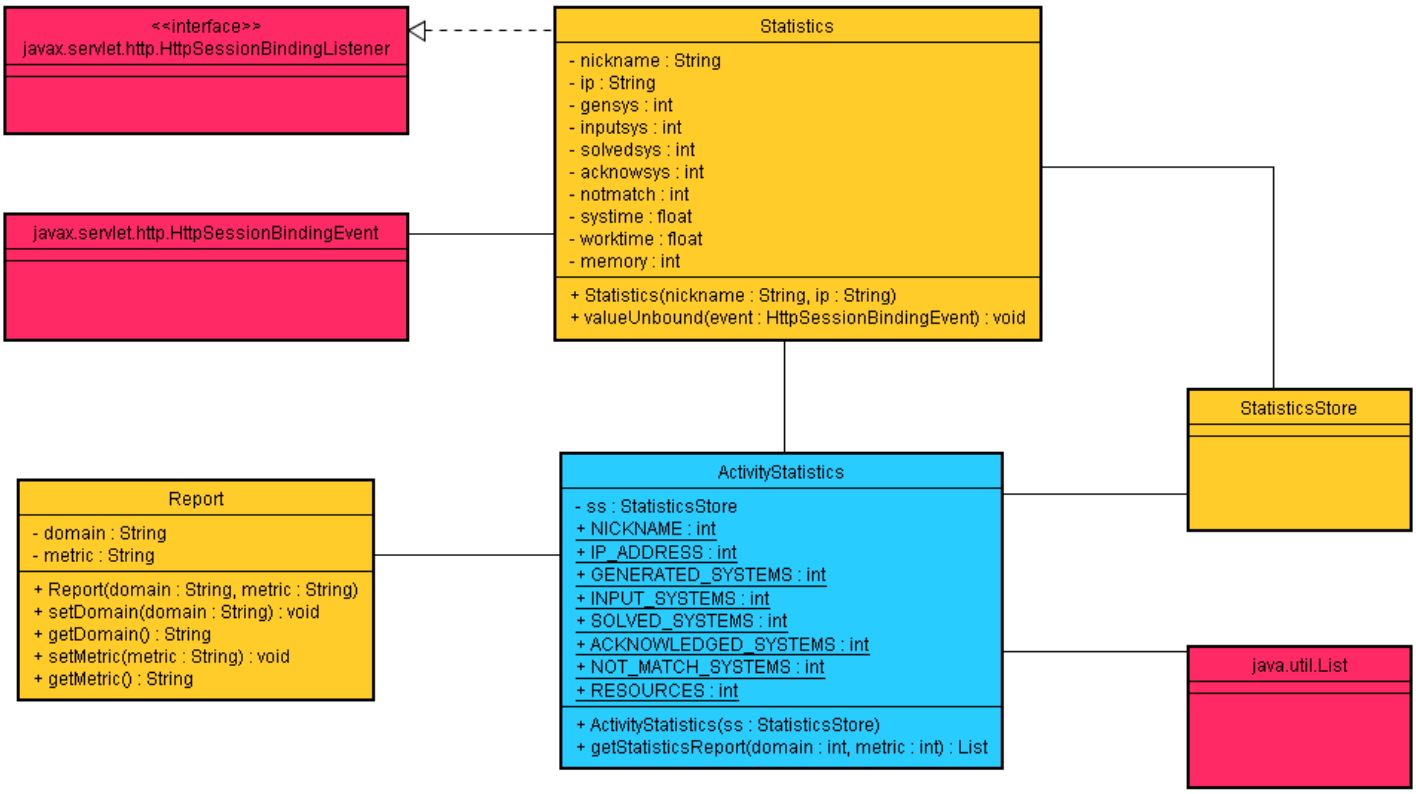

Figure 2: Activity statistics