Architecture Design

For KDD- Service based Numerical Entity Searcher

(KSNES)

Version 2.1

Submitted in partial fulfillment of the Masters of Software Engineering degree.

Naga Sowjanya Karumuri
CIS 895 – MSE Project
Department of Computing and Information Sciences
Kansas State University

Committee Members
Dr. William. H. Hsu
Dr. Torben Amtoft
Dr. Mitchell Neilsen
<table>
<thead>
<tr>
<th>Version #</th>
<th>Changed By</th>
<th>Release Date</th>
<th>Change Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version 1.0</td>
<td>Sowjanya</td>
<td>03/19/09</td>
<td>Initial Release</td>
</tr>
<tr>
<td>Version 1.1</td>
<td>Sowjanya</td>
<td>03/23/09</td>
<td>Updated Formal Specification and added class descriptions</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1: Package View ................................................................. 6
Figure 2: NPE Class Diagram ....................................................... 6
Figure 3: NDPR Class Diagram .................................................... 7
Figure 4: Sequence Diagram ....................................................... 9
Table of Contents

Change Log .................................................................................................................................................2

List of Figures ...............................................................................................................................................3

1. Introduction ...........................................................................................................................................5
   1.1. Background ...................................................................................................................................5
   1.2. References .......................................................................................................................................5

2. KDD- Service based Numerical Entity Searcher .................................................................................5
   2.1. Package View ...................................................................................................................................5
   2.2. Numerical Phrase Extractor Package .............................................................................................6
       2.2.1. Class Description .......................................................................................................................7
   2.3. Number-Unit/Date Pattern Recognizer Package ...........................................................................7
       2.3.1. Class Description .......................................................................................................................8
   2.4. Sequence Diagrams ......................................................................................................................8
       2.4.1. User Interacting with the System .................................................................................................8

3. Formal Specification ...............................................................................................................................9
1. Introduction

The purpose of this document is to provide the architectural design of KSNES. The documentation presents the class and sequence diagrams and explanation is provided for each diagram. The last section has the formal specification of the Numerical Phrase Extractor and the Number-Unit/Date Pattern Recognizer.

1.1. Background

The purpose of the project is to extract the numerical information from the given raw text and present the values, units and the numerical phrase type. The whole system is set as a service on the server and a webpage is created for the user to enter the raw text and view the required information.

1.2. References


2. KDD- Service based Numerical Phrase Recognizer

2.1. Package View

As discussed in the Vision document, the present system has client server architecture. The webpage acts as the client and the KSNES system acts as the server. In this document, details of Numerical Phrase Extractor and Number-Unit/Date Pattern Recognizer are presented since the developer has worked on these modules and the POS Tagger is taken from a third party and hence not documented.

In the below package view, we see the three main modules of the server as three packages that work together in sequence to process the raw text and extract the numerical phrases from it.
We see that Number-Unit/Date Phrase Recognizer depends on Numerical Phrase Extractor and this in turn depends on POS Tagger.

Figure 1: Package View

2.2. Numerical Phrase Extractor Package

Figure 2: NPE Class Diagram
2.2.1. Class Description

- **PatternMatch**: This class starts the NPE package working by calling a method from MyFunctions to start reading the given tagged input file.
- **MyFunctions**: This class holds the functions to read from the input file, writes the map values to the output file, delete the files and parse the line with the patterns available in the MyPatterns class.
- **MyPatterns**: This class contains all the patterns and each pattern takes the input line and when found any match, is written to the map. This map is printed back to the file by calling a function from MyFunctions class.

2.3. Number-Unit/Date Phrase Recognizer Package

![Figure 3: NDPR Class Diagram](image)
2.3.1. Class Description

- **Main**: This is the class that reads the input from the file and calls the number or the date functions based on the type of the input.
- **Number**: This class parses the given number phrase and evaluates the value, unit, unit-type and the bound corresponding to that phrase.
- **Date**: This class parses the given date phrase and prints the date as mm/dd/yyyy. If there is any second date it is also parsed as mm/dd/yyyy.
- **Quantity**: This class has the basic functions defined that are used by the number and the date classes.

2.4. Sequence Diagrams

The system is more like a black box to the user and hence the user has very less to interact with the system but the below diagram explains how the user system interaction is taking place. The message passing between the user and the system are pseudo as it has been set by one of the KDD members. Hence the below sequence diagram explains the process happening between the user and the system but do not present the actual methods that have been invoked during the flow.

2.4.1. User Interacting with the System

**Prerequisites:**
- KSNES is set as a service on the server.
- A webpage is designed for the user to access the system through it.

**Sequence of Events:**
1. User opens the browser
2. User opens the KSNES webpage
3. User enters the raw text into the text box in the page
4. User clicks the send button
5. User sees a new webpage displaying the given text, chunks, chunk types, values, units, unit-types and bounds.
6. User may close the browser or enter a new raw text again.

**Post-Conditions:** User sees a new webpage displaying the given text, chunks, chunk types, values, units, unit-types and bounds.
3. Formal Specification

```plaintext
model KSNES

--
-- NPE PACKAGE
--
```
class Readfile
attributes
  rline: String
end

class Writefile
attributes
  wline: String
end

class Map
attributes
  linenum: Integer
  value: String
end

class Datetemp
end

class PatternMatch
attributes
  mfunction: MyFunctions
end

class PatternMatch
attributes
  mfunction: MyFunctions
end

class MyFunctions
attributes
  tmp: String
  iline: String
  fchunk: String
  passon: String
  rf: Readfile
  wf: Writefile
  mchunkorder: Map
operations
  startparsing()
postchunks()
deletefile()
cleanchunk(ichunk: String)
parseline(taggedline: String)
end

class MyPatterns
attributes
  chunk: String
  mf: MyFunctions
  chunkorder: Map
operations
  p_abtfrac(tline: String): String
  p_age(tline: String): String
  p_alphanum(tline: String): String
  p_ampm(tline: String): String
  p_and(tline: String): String
  p_anydate(tline: String): String
  p_btwfrm(tline: String): String
  p_btwfrmd(tline: String): String
  p_date(tline: String): String
  p_days(tline: String): String
  p_centuary(tline: String): String
  p_hyphenww(tline: String): String
  p_hyphennumnum(tline: String): String
  p_in(tline: String): String
  p_mids(tline: String): String
  p_months(tline: String): String
  p_numunit(tline: String): String
  p_per(tline: String): String
  p_percentinches(tline: String): String
  p_range(tline: String): String
  p_ratio(tline: String): String
  p_tty(tline: String): String
  p_twmy(tline: String): String
  p_xbits(tline: String): String
  p_yrange(tline: String): String
end

--
-- NDPR PACKAGE
class main
attributes
    infile: Readfile
operations
    batch_process(infile: String)
    main(argc: Integer, argv: Set(String))
end

class number
attributes
    unit: String
    cat: String
    value: Integer
    inputnum: String
operations
    get_units()
    is_equal(number: Integer): Boolean
    print()
end

class date
attributes
    month: Integer
    day: Integer
    year: Integer
    format: String
    modifier: String
    second_date: Datetemp
    inputdate: String
operations
    get_modifier()
    make_relative_date(dt:Integer)
    is_equal()
    print()
end

class quantity
attributes
    dbl2string: String
int2string: String
type: String
phrase: String
bound: String
operations
  match(s:String, pattern:String, v:Set(String))
  subst(s:String, pattern:String, replace:String)
  split(s:String, delim:String): Set(String)
  convert_to_integer(phrase:String):String
  get_bound()
  is_equal()
end

--
-- ASSOCIATIONS
--

-- This association will not appear in the class diagram but
-- is seen as the relation between the packages

association Packagerelations between
  PatternMatch[1] role usedby
    main[1] role connectsto
end

--
-- NPE PACKAGE
--

association PatternUseFunctions between
  PatternMatch[1] role calledbypatternmatch
    MyFunctions[1] role maincallsfuncs
end

association FunctionsCallPatterns between
  MyFunctions[1] role patternSCALLFUNCTIONS
    MyPatterns[1] role functionscallpatterns
end
association InputMaybeNumber between
    main[1] role inputmaybenumber
    number[1] role maincallsnum
end

association InputMaybeDate between
    main[1] role inputmaybedate
    date[1] role maincallsdate
end

association NumberUseQuantity between
    number[1] role quantitybynum
    quantity[1] role numberusequantity
end

association DateUseQuantity between
    date[1] role quantitybydate
    quantity[1] role dateusequantity
end

association NumberUseUnitslist between
    number[1] role maybecontainedby
    unitslist[1] role containsunit
end

constraints

context PatternMatch
inv passonfile:
    maincallsfuncs.rf = connectsto.infile

--
-- The input to the NDPR Package is either a number or a date
--

context main
inv dateornumber:
   ((infile.rline = maincallsnum.inputnum) or
    (infile.rline = maincallsdate.inputdate)) and
    (maincallsnum.inputnum <>
    maincallsdate.inputdate)

--
-- The outputs written to the output file are those put in the
map after processing
-- the input file
--

context MyFunctions
inv printcorrectoutput:
   mchunkorder=functionscallpatterns.chunkorder
    and wf.wline = mchunkorder.value

--
-- If the unit is not mentioned in the list then the unit is -
-- misc
--

context number
inv unitmatch:
   (containsunit.lunit = unit) or unit = 'Misc'