Format processing in telematic applications

Academic Year: (2020 / 2021) **Review date: 05-07-2020**

Department assigned to the subject: Telematic Engineering Department

Coordinating teacher: SANCHEZ FERNANDEZ, LUIS

Type: Compulsory ECTS Credits : 6.0

Year : 3 Semester : 2

REQUIREMENTS (SUBJECTS THAT ARE ASSUMED TO BE KNOWN)

- Programming
- Systems Programming

OBJECTIVES

The goal of this subject is that the students acquire the basic knowledge, competences and skills required to process documents exchanged in telematic applications. Both the traditional approach (language processing) and the XML approach are studied. The relation between XML document processing and classical language processing will also be discussed.

With respect to the knowledge the students should learn, at the end of the course, the student will be able to:

- Know the definitions of alphabet, string and language.
- Understand the parts in which a compiler is divided, the tasks that each part performs and the relations between the different parts.
- Know some formalisms to define languages: regular expressions and grammars
- Know the role of a lexer and how to specify it.
- Know the role of a parser and how to specify it.
- Know the how an LR(k) parser works.
- Know the definitions of attribute and synthesized and inherited attribute.
- Know how to specify an attribute.
- Know the XML language.
- Know the DTD format to define XML applications.
- Know the environment to process XML documents DOM.
- Know the stylesheet language XSLT.

With respect to specific instrumental competences, the students should be able to:

- Understand which is the language defined by a given regular expression or grammar.
- Write the specification for the lexer and the parser of a given language.
- Program the required Java classes that represent the abstract syntax trees corresponding to strings of a given language.
- Use analyzer generators.
- Implement a symbol table.
- Implement a synthesized or an inherited attribute, given its specification.
- Implement a type checker.
- Implement other checkings required by a semantic analyzer.
- Design and implement a code generator in Java.
- Decide whether an XML document is valid according to a given DTD.
- Write simple programs that process XML documents.
- Write simple XSLT stylesheets.

With respect to general competences, the student should show ability to:

- identify, formulate and solve engineering problems. (PO: e)
- use the techniques, skills and modern tools necessary for engineering practice. (PO: k)

DESCRIPTION OF CONTENTS: PROGRAMME

1. Definition, analysis and processing of generic formats:
   - Definition of alphabet, string and language
   - Grammars and regular expressions
   - Compiler phases
   - Lexical and sintax analysis
   - Analyzer generators
   - Semantic analysis and code generation
2. XML as an internal, exchange and presentation format:
   - The XML language
   - DTD
   - Document processing with DOM
   - XPath
   - XSLT

LEARNING ACTIVITIES AND METHODOLOGY

The activities that will be carried out in this subject are:

- Lectures, in which the lecturer will present the main concepts studied in the subject. In this type of sessions interactivity and discussion of problems introduced by the lecturer will be encouraged.
- Practice sessions, in which real problems are proposed and the students will have to analyze and propose possible solutions to them.
- Labs: in these sessions the students should develop a simple compiler as well as small developments with XML technology. With the supervision of a teacher, the students should procede to the implementation of the required programs. The students should have to do an additional work after each lab session, for several days, to complete the assignment. During these days, the students can ask questions to the teaching team through different channels.

ASSESSMENT SYSTEM

- One partial exam on lexical and syntax analysis composed on theoretic questions and/or problems (weight 30 points)

- One partial exam on semantic analysis and code generation composed on theoretic questions and/or problems (weight 20 points)

- Final exam on XML (weight 30 points).

- Evaluation of the assignments the students have done in the labs (weight 10 points).

- Lab exam (weight 10 points).

Improvement

Students that simultaneusly obtain an score of at least 5 points in the evaluation of the lab assignments, an score of at least 5 points in the lab exam and for which the sum of the scores of the two partial exams is greater than 20 points will be graded in the continuous evaluation, in case that would be favourable to the student, as follows.

Being:

NP: sum of the scores of the two partial exams (weight 50 points)
NL: socore of the lab assignments (weight 10 points)
NEL: score of the lab exam (weight 10 points)
NEC: total score in the continuous evaluation

- If 20 < NP <= 30

NEC=NP + NL + NEL + (NL -NP/5) * (NP-20)/10

- If 30 < NP <= 45

NEC= NP * 4/5 + NEL + NL * (2 + (NP-30)/15)

- If NP > 45

NEC= NP * 4/5 + NEL + NL * 3

Requirements to pass

Students that follow the continuous evaluation should fulfill the following requirements to pass the subject:

1) The sum of the scores of the continuous evaluation and the final exam must be at least 50 points.
2) The score of the lab assignments must be at least 5 points.
3) The score of the final exam must be at least 12 points.

% end-of-term-examination:	30
% of continuous assessment (assigments, laboratory, practicals…):	70

BASIC BIBLIOGRAPHY

- Scott E. Hudson, modificado por Frank Flannery, C. Scott Ananian y Dan Wang, guiados por Andrew W. Appel CUP User's Manual, http://www.cs.princeton.edu/~appel/modern/java/CUP/manual.html, 1999
- A.V. Aho, R. Sethi, J.D. Ullman Compilers: principles, techniques and tools, Pearson Education.
- Andrew W. Appel Modern Compiler Implementation in Java, Cambridge University Press.
- Atul Kahate XML & Related Technologies, Pearson Education, 2009
- Carlos Delgado Kloos, Peter T. Breuer Definición de lenguajes de programación, http://www.it.uc3m.es/luis/fo1/es/parte1.pdf.
- E.R. Harold, W.S. Means XML in a Nutshell, Third edition, O'Reilly.
- Elliot Berk JLex: A lexical analyzer generator for Java(TM), http://www.cs.princeton.edu/~appel/modern/java/JLex/current/manual.html, 2000
- Erik T. Ray Learning XML, Second Edition, O'Reilly, 2003
- Evan Lenz XSLT 1.0 Pocket Reference, O'Reilly, 2005
- Gregorio Fernández Fernández, Fernando Sáez Vacas Fundamentos de informática: [lógica, autómatas, algoritmos y lenguajes], Anaya Multimedia.
- Luis Sánchez Fernández Introduction to Compiler Development with Java, Apuntes de la asignatura.
- World Wide Web Consortium XML Technology. http://www.w3.org/standards/xml/, World Wide Web Consortium.