ProWF: Workflow Editor

Architecture Notebook

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Purpose

Dokumentu honek deskribatu egiten ditu proiektuan zehar, eta zehazki Workflow Editor azpisistemaren atalean, filosofia, erabakiak, baldintzak, justifikazioak eta garrantzizkoa izan daitekeen edozer.

# Architectural goals and philosophy

Workflow Editor azpisistemaren arkitektura finkoa izan behar da, nahiz eta berak erabiliko duen workflow-lengoaia aldatu, azpisistema ez da aldatuko.

Azpisistemak hurrengo helburu nagusiak ditu:

* Workflow-lengoaia bat definituta izan behar du, lengoaia horrek workflow-ereduak exekutatzeko prest egon behar du.
* Eraldaketa-prozesu baten bitartez sortutako workflow-ereduak inferentzia motor baten bidez exekutatu behar dira.
* Workflow-ereduen informazioa datu-base erlazionalean gorde behar inferentzia motorraren bidez.

# Assumptions and dependencies

Hauek dira Workflow Editor azpisistemak arkitekturalki dituen dependentziak:

* GraphViz, Gephi, Protégé eta EHSIS teknologien menpe dago, menpekotasun teknologikoa sortuz.
* Notepadd++ bezalako testu-editore bat erabiltza beharrezkoa da workflow-ereduak sortzeko.

# Architecturally significant requirements

Sistemaren betekizunak “Sistemaren Espezifikazioa” ataleko “Betebeharren Espezifikazioa” dokumentuan definitu dira.

# Decisions, constraints, and justifications

* Azpisistema guztiz independentea izan behar da IO-System azpisistemarekiko. Bigarrenak, ostera, lehenengoaren existentzia behar du funtzionatzeko.
* Workflow-ereduak sortzeko testu-editorea erabiltzea erabaki da. Editore grafiko bat inplementatzeak denbora-kostu handiak ekarriko baitzuen.
* GraphViz, Gephi eta Protégé softwareak erabiltzea erabaki da eraldaketa-prozesua egiteko. Izan ere, DOT lengoaiatik zuzenean EHSIS inguruneak exekutatzeko prest egoteko metaeredu bat definitzeak denbora-kostu handiak ekarriko baitzuen, baina egokiena meteredua definitzea izango litzateke.
* Workflow-ereduen irudia sortzean, nabigagarria izango da. Horrela, workflowaren garbitasuna asko handituko da eta ulergarriagoa izango da.

# Architectural Mechanisms

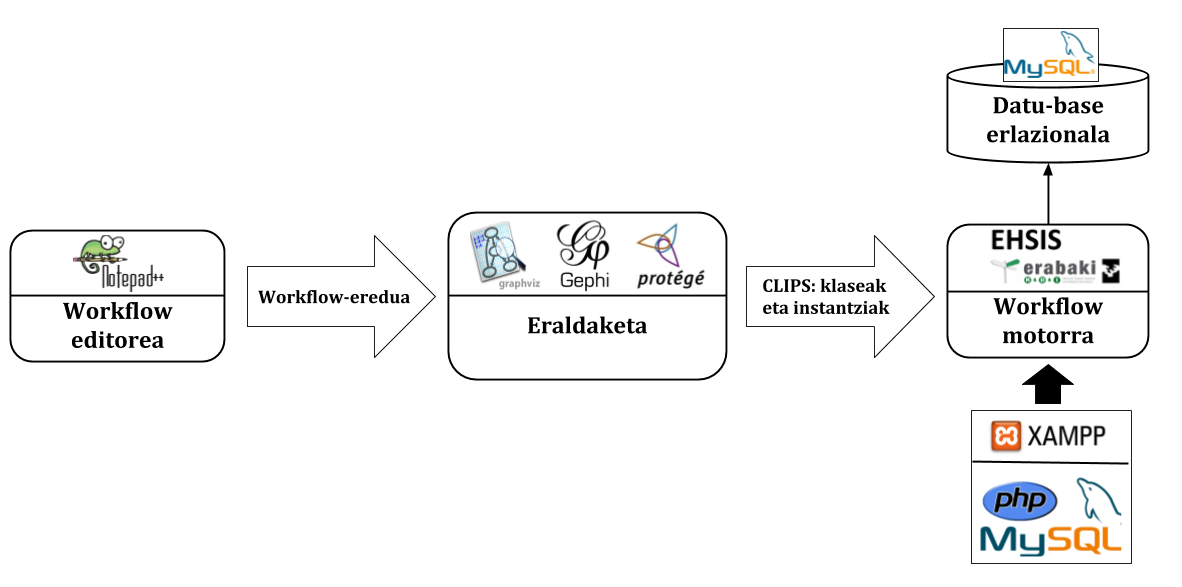
## Workflow-eredutik irudira

Graphviz softwarearen bitartez, DOT lengoaian idatzitako workflow-ereduak irudi bihurtuko dira. Irudia nabigagarria izango da. Irudia nabigatzaile baten zabaltzeko prestatuta egongo da.

## Workflow-eredutik datu-base erlazionalera

Eraldaketa-prozesuaren bitartez, inferentzia motorrarekin workflow-ereduetan idatzitako informazioa datu-base erlazionalean gordeko da.

# Layers or architectural framwork



Jarraian, arkitekturaren osagai eta elementu guztiak deskribatuko dira:

* Workflow editorea: definitutako lengoaia grafiko bat erabilita, testu editore batekin, DOT lengoaiaren bitartez *workflow*-ereduak sortzeko balio du. “*Workflow-lengoaiaren Eskuliburua*” dokumentuan jarri dira lengoaia grafikoaren xehetasunak.
* Eraldaketa-prozesua: *DOT* lengoaian idatzitako *workflow*-eredua *CLIPS* lengoaiako klase eta instantzietan eraldatzea du helburu. Horretarako, “*Workflow Editor - Eskuliburua*” dokumentu jarraitu behar da.
* Datu-base erlazionala: sortutako *workflow*-ereduen informazio gordetzeaz arduratzen da.
* *Workflow* motorra: *CLIPS* lengoaiara eraldatutako *workflow*-ereduaren klaseak eta instantziak baliatuz, *workflowa* exekutatzeko erregela batzuen bidez, informazio guztia datu-base erlazionalean gordeko du.