**Documentul de Proiectare a Solu**t**iei Aplica**t**iei Software**

**(Software Design Document)**

Versiunea 1.0

28 Octombrie 2012

**Preprocessing GUI**

Echipa Blue, Facultatea de Automatica si Calculatoare, Universitatea Politehnica, Bucuresti

Cuprins

1. **Scopul documentului…………………………………………..3**
2. **Obiective………………………………………………………..3**
3. **Continutul documentului……………………………………...4**
4. **Modelul datelor………………………………………………...5**
   1. Structuri de date globale**…………………………………..5**
   2. Structuri de date de legatura**……………………………....5**
   3. Structuri de date temporare**………………………………..6**
   4. Formatul fisierelor utilizate**………………………………..6**
5. **Modelul arhitectural si modelul componentelor……………...6**
   1. Arhitectura sistemului**……………………………………..6**
      1. Sabloane arhitecturale folosite**……………………...6**
      2. Diagrama de arhitectura**…………………………….7**
   2. Descrierea componentelor**…………………………………8**
   3. Restrictii de implementare**…………………………………8**
   4. Interactiunea dintre componente**…………………………...9**
6. **Modelul interfetei cu utilizatorul……………………………...10**
   1. Succesiunea interfetelor**…………………………………...10**
   2. Ferestrele aplicatiei**………………………………………..11**
7. **Elemente de testare……………………………………………..12**
   1. Componente critice**………………………………………..12**
   2. Alternative**………………………………………………....13**
8. Scopul documentului

Acest document are ca scop descrierea intr-un mod realist si cat mai complet a interfetei necesare procesarii de imagini corespunzatoare proiectului Preprocessing GUI.

Documentul serveste drept ghid in construirea solutiei pentru echipa de dezvoltare.

1. Obiective

In urma realizarii proiectului vom putea sa:

* stabilim parametrii necesari preprocesarii
* evaluam rezultatele diferitelor preprocesari asupra imaginilor;
* cream un xml valid si corect corespunzator xsd-ului pus la dispozitie

1. Continutul documentului

Documentul este impartit in 4 sectiuni dupa cum urmeaza:

* **Modelul datelor** in care se prezinta
  + principalele structuri de date folosite
  + formatul fisierelor utilizate
* **Modelul arhitectural si modelul componentelor**
  + prezinta arhitectura sistemului
  + descrie componentele sistemului, restrictiile de implementare si interactiunea dintre ele
* **Modelul interfetei cu utilizatorul** prezinta
  + ferestrele aplicatiei
  + succesiunea ferestrelor in cadrul sistemului
* **Elemente de testare**
  + identifica componentele critice
  + propune alternative de proiectare ale acestora

1. Modelul datelor

# . Structuri de date globale

Structura de date globala corespunzatoare aplicatiei noastre este reprezentata de clasa Main in care va fi afisata fereastra principala a aplicatiei. Aici utilizatorul va descoperi o diversitate de filtre pe care le poate aplica unei imagini selectate.

Tot prin intermediul clasei Main celelalte clase isi vor putea accesa reciproc structurile de date interne.

# . Structuri de date de legatura

Modulul GUI va comunica cu modulul XSD in urma caruia se vor stabili parametrii necesari pentru diferite procesari ale imaginilor. Totodata se vor cunoaste numarul de parametrii pe care utilizatorul trebuie sa-i specifice pentru a obtine rezultatul asteptat.

Modulul GUI comunica apoi cu modulul LOGIC caruia ii va trimite atat parametrii obtinuti din xsd cat si parametrii transmisi de utilizator si se va genera xml pe baza datelor furnizate.

Modulul XML va comunica cu modulul XSD pentru validarea corectitudinii informatiilor obtinute in xml-ul generat la etapa anterioara. In cazul in care xml-ul corespunde cu cerintele si restrictiile specificate in xsd va fi in varianta finala. In caz contrar se va reveni la comunicatia dintre modelele GUI si LOGIC pentru generarea unui nou xml.

# . Structuri de date temporare

Pe langa principalele structuri de date aplicatia va mai avea o serie de date al caror ciclu de viata fie se limiteaza la corpul unei functii fie la transmisia de informatii intre module.

# 4.4. Formatul fisierelor utilizate

Pentru introducerea de informatii in vederea procesarii imaginilor se vor folosi fisiere XSD ce vor fi parsate. Se vor obtine astfel parametrii necesari tipului de procesare selectat si se va genera un fisier xml care sa corespunda schemei de definitie precizata in xsd.

1. Modelul arhitectural si modelul componentelor

# . Arhitectura sistemului

* + 1. **Sabloane arhitecturale folosite**

Solutia software a proiectului se bazeaza pe sablonul de *event-driven architecture.*

Acest sablon promoveaza producerea, detectarea si reactia la evenimente. In cazul aplicatiei noastre evenimentele vor fi generate prin selectarea unuia din filtrele de procesare disponibile aplicatiei si rularea acestuia.

* + 1. **Diagrama de arhitectura**

Diagrama de arhitectura prezentata mai jos ilustreaza componentele aplicatiei si modul in care acestea interactioneaza pentru obtinerea rezultatului dorit.

**GUI**

**XSD**

**LOGIC**

**XSD**

**XMLi**

**XMLf**

# . Descrierea componentelor

Aplicatia prezinta urmatoarele module:

* **Modulul GUI ( Interfata grafica)** se ocupa de desenarea si afisarea ferestrei principale cat si a ferestrelor intermediare de preluare a datelor de intrare de la utilizator.
* **Modulul de Logica (Business Logic)** se ocupa de coordonarea celorlalte module astfel incat rezultatele rularii lor sa conduca la obtinerea unui fisier xml conform cu specificatiile din fisierul xsd.
* **Modul de procesare XSD** se ocupa de parsarea fisierelor xsd cu scopul de a stabili parametrii necesari unui anumit filtru.
* **Modul de validare XML** se ocupa cu validarea xml-ului obtinut in urma determinarii tuturor parametrilor necesari si filtrului pe o imagine introdusa de utilizator astfel incat sa fie corespunzator restrictiilor din xsd.

# . Restrictii de implementare

Modulele aplicatiei trebuie sa respecte urmatoarele restrictii de implementare:

* Modulul de validare XML va fi dezvoltat pentru a suporta tehnologiile XML:XSL si XSD
* Modulul de Logica va fi dezvoltat pentru a genera un fisier xml corespunzator parametrilor determinati si filtrului aplicat.
* Toate modulele aplicatiei vor fi dezvoltate in limbajul de programare *JAVA Standard Edtition Versiunea 6.*

# . Interactiunea dintre componente

In momentul pornirii aplicatiei de catre utilizator este apelata clasa Main din Modulul de Logica. Aceasta va afisa fereastra principala a aplicatiei, corespunzatoare modulului de GUI.

In acest moment utilizatorul introduce in interfata imaginea pe care doreste sa o proceseze si poate selecta filtrele pe care i le va aplica. Este apelat in continuare Modulul de procesare XSD in care se determina numele imaginii si numele si tipul parametrilor necesari filtrului dorit. Acestia vor fi introdusi de catre utilizator in interfata grafica. Dupa rularea aplicatiei se va obtine un fisier xml care trebuie sa corespunda standardelor si restrictiilor precizate in fisierul xsd.

Deoarece fisierul xsd poate fi updatat se va face validarea fisierului XML in cadrul modulului de Validare. In cazul in care nu corespunde se va reveni la modulul de Logica si va fi parsat noul fisier XSD, reluandu-se algoritmul pana la obtinerea unui fisier xml corect si valid.

Desenare GUI

**Modul GUI**

**Modul Logica**

Validare XML

Generare XML

Alegere filtru

Feedback

Feedback

**validateXml**

**Modul Validare XML**

Feedback

Modul Procesare XSD

**parseXsd**

Utilizatorul deschide aplicatia

**drawWindow**

**Main Class**

1. Modelul interfetei cu utilizatorul

# 6.1. Succesiunea interfetelor

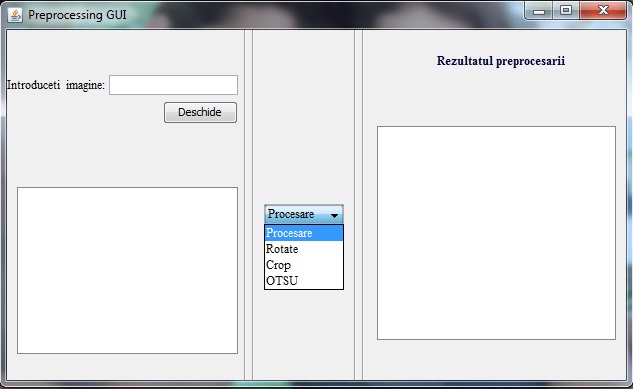
In cadrul modulului GUI, ferestrele sunt afisate repectand un flux bine stabilit.

La pornirea aplicatiei apare fereastra principala care este impartita in 2 Panel-uri. In panel-ul din stanga avem un textfield pentru selectarea imaginii si o zona de afisare a imaginii initiale. Panel-ul din dreapta va afisa doar imaginea obtinuta prin procesare. Intre cele 2 panel-uri avem un meniu ce ne permite sa ne alegem filtrul pe care dorim sa-l aplicam.

La selectarea unuia din filtre se va deschide automat fereastra corespunzatoare filtrului in care vom avea un numar de textfield-uri egal cu numarul de parametrii necesari filtrului pentru ca utilizatorul sa-i specifice.

# 6.2. Ferestrele aplicatiei

* Fereastra principala a aplicatiei arata astfel:



Fereastra prezinta 2 panel-uri: cel din stanga permite utilizatorului introducerea unei imagini si afisarea in spatiul alb a acesteia. Panelul din dreapta va arata imaginea obtinuta in urma preprocesarii, a aplicarii unuia sau mai multor filtre.

In partea din mijloc avem un meniu din care ne putem alege unul din cele 3 filtre pe care le vom aplica imaginii initiale.

* Ferestrele asociate celor 3 filtre vor contine in partea de sus numele Filtrului, numele imaginii si o serie de textfield-uri corespunzatoare parametrilor pe care utilizatorul ii va introduce de la tastatura.

1. Elemente de testare

# Componente critice

Performanta aplicatiei este direct implicata de performantele modulelor Logica si Parsare Xsd si de legatura dintre cele 2. Viteza de parsare a documentelor xsd, transmisia parametrilor necesari procesari, viteza de procesare a imaginii si de crearea a fisierului xml final au un rol decisiv in performanta generala a sistemului.

# Alternative

In vederea scaderii timpului de parsare a fisierelor xsd, vom crea un parser care va stabili automat parametrii ce vor fi cititi si restul informatiilor necesare aplicarii filtrelor.