Document de Proiectare a Soluției Aplicației Software

(Software Design Document)

Layout Analysis

Echipa "Echipa T4 de miercuri de la 8 la 10"

Facultatea de Automaticț și Calculatoare, Universitatea Politehnică București

Versiunea 1.0

23 Octombrie, 2012

Cuprins

|  |  |
| --- | --- |
| Cuprins | 2 |
| 1. Scopul documentului | 3 |
| 2. Continutul documentului | 3 |
| 3. Modelul datelor | 3 |
| 3.1. Structuri de date globale | 3 |
| 3.2. Structuri de date de legatura | 4 |
| 3.3. Structuri de date temporare | 4 |
| 3.4. Formatul fisierelor utilizate | 4 |
| 4. Modelul arhitectural si modelul componentelor | 5 |
| 4.1. Arhitectura sistemului | 5 |
| 4.1.1. Sabloane arhitecturale folosite | 5 |
| 4.1.2.1. Diagrama de arhitectura | 6 |
| 4.1.2.2. Detalierea arhitecturii componentelor | 7 |
| 4.2. Descrierea componentelor | 8 |
| 4.3. Restrictiile de implementare | 8 |
| 4.4. Interactiunea dintre componente | 9 |
| 5. Modelul interfetei cu utilizatorul | 9 |
| 5.1. Succesiunea interfetelor | 9 |
| 5.2. Ferestrele aplicatiei | 11 |
| 5.2.1. Fereastra de selectare a documentului | 11 |
| 5.2.2. Editarea caracterelor în interiorul liniilor | 12 |
| 5.2.3. Editarea liniilor în paragrafe | 12 |
| 5.2.4. Editarea paragrafelor în document | 14 |
| 5.2.5. Ecranul de paginare și generare | 15 |
| 6. Elemente de testare | 16 |
| 6.1.Componente critice | 16 |

1. Scopul documentului

Acest document are ca scop descrierea și arhitecturarea soluției software intitulată "Layout Analysis" cu scopul de a corecta erorile din gruparea caracterelor în rânduri și a rândurilor în blocuri, paginării.

Documentul descrie clar, complet și concis obiectivele proiectului, stând la baza contractului de execuție a proiectul și servind drept ghid unic de proiectare și implementare a soluției pentru echipa de dezvoltatori.

1. Conținutul documentului

Secțiunile principale ale documentului sunt cele ce urmeaza:

* Modelul datelor - reprezintă principalele structuri de date folosite.
* Modelul arhitectural și al componentelor - reprezintă arhitectura sistemului, descrie modulele componente și afisează diferite șabloane arhitecutale și/sau de programare.
* Modelul interfeței cu utilizatorul - prezintă modul prin care un utilizator interactionează cu aplicația precum și succesiunea ferestrelor in fluxul de aplicație.
* Modelul de testare - prezintă componentele cu potențial mare de eșec și alternative de proiectare a acestora.

3. Modelul datelor

3.1. Structuri de date globale

Structura globală de date este un arbore, reprezentând o abstractizare optimă a organizării și a conținutului fișierului .xml de ieșire, conform cu schema aferentă. Astfel, acesta va fi format din mai multe tipuri de noduri:

* *DocumentNode* - nodul corespunzător tag-ului <Document> (rădăcina .xml-ului)
* *ComposedBlockNode* - noduri corespunzătoare tag-urilor <ComposedBlock> ; fiecare nod are ca părinte rădăcina
* *TextBlockNode* - noduri corespunzătoare tag-urilor <TextBlock> ; fiecare nod are ca părinte fie rădăcina, fie un nod de tip ComposedBlockNode
* *ImageBlockNode* - noduri corespunzătoare tag-urilor <ImageBlock> ; fiecare nod are ca părinte rădăcina
* *TextLineNode* - noduri corespunzătoare tag-urilor <TextLine> ; fiecare nod are ca părinte un nod de tip TextBlockNode
* *StringNode* - noduri corespunzătoare tag-urilor <String> ; fiecare nod are ca părinte un nod de tip TextLineNode.

Rădăcina arborelui este unică și este întotdeauna un nod de tip *DocumentNode*.

Frunzele arborelui (nodurile terminale) sunt fie noduri de tip *ImageBlockNode*, fie *StringNode* și conțin datele corespunzătoare unei imagini, așa cum au fost specificate în fișierul .xml de intrare, respectiv textul parsat și eventual corectat de către utilizator.

3.2. Structuri de date de legătură

Legătura dintre reprezentarea internă, arborescentă, a datelor și cea vizuală, care ține de modulul GUI, se va face prin intermediul câmpurilor de text afișate pe ecran (editabile sau needitabile de către utilizator) și a formelor geometrice (chenare / highlight-ing) ce delimitează zonele de text. Din modulul GUI se vor transmite modulului de logică date precum coordonatele selecției cu mouse-ul, indexul blocului selectat, indexul unei linii marcată pentru eliminare etc. Modulul de logică transmite modulului de procesare XML ierarhia datelor, precum și datele propriu-zise. De asemenea, modulul de procesare XML va comunica și cu modulul GUI în situația în care utilizatorul încarcă pentru editare un fișier .xml; în acest caz, se vor transmite atât datele efective - textul - cât și coordonatele acestuia în pagină și organizarea pe blocuri.

3.3. Structuri de date temporare

Structurile de date temporare utilizate vor avea rol strict auxiliar și nu vor consuma resurse semnificative de memorie și/sau efort computațional.

3.4. Formatul fișierelor utilizate

Fișierele de intrare pot fi:

* documente .xml valide și bine formate, conforme cu schema .xsd prestabilită;
* imagini în format .tiff îndreptate în așa fel încât textul să fie orizontal și cu contrastul ajustat în așa fel încât textul să fie ușor lizibil.

Fișierele de ieșire sunt documente .xml valide și bine formate, conforme cu schema .xsd prestabilită.

4. Modelul arhitectural şi modelul componentelor

4.1. Arhitectura sistemului

4.1.1. Şabloane arhitecturale folosite

Componenta de ”Layout Analysis” a proiectului global respectă o arhitectură de tip *”batch processing”*.

Datele primite la intrare trec printr-o serie de etape disjuncte până să ajungă în starea de date valide de ieșire.

Fiecare etapă a procesării are atribuit un modul aplicație care respectă un protocol aplicație concretizat printr-o serie de interfețe de programare și date, încât imlementarea să poată varia independent de interfată (conform principiului de *”black box”*).

O astfel de arhitectură oferă următoarele avantaje:

* cuplare redusă intre componentele aplicației
* ușurință în implementarea unor noi elemente componente și intercalării lor datorită programării orientate pe interfețe
* exitindere ușoară către arhitecturi de tip cluster ori multiprocesor datorită dependențelor de date reduse impuse de arhitectură
* arhitectură ușor inteligibilă, simplă, deci cu o curbă de învățare nesemnificativă

4.1.2.1. Diagrama arhitecturală

Diagrama de arhitectură de mai jos descrie componentele arhitecturii aplicaţiei şi relaţiile de

interacţiune dintre acestea.

Image to XML converter

Layout Editing & Content Overriding

XML

Image

Input XML

Output XML

4.1.2.2. Detalierea arhitecturii componentelor

Componenta de *”Layout Synthesis”.*

Component detector

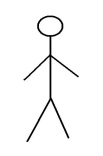
OCR

Image to XML converter

Imagine

XML Intermediar

Componenta de ”Layout and Content Editing”



User Interaction

Layout Editing & Content Overriding

XML

Output XML

4.2. Descrierea componentelor

Aplicația cuprinde următoarele componente:

* *”Image to XML Converter”* (Convertor imagine - XML)

Componenta primește input o imagine și are rolul de a efectua o analiză structurală în vederea generării unui fișer XML conform specificației XSD.

Există doua funcții majore implementate la nivelul aceste componente:

* + - Recunoașterea caracterelor existente in imagine (OCR)
    - Gruparea acestora in structuri de layout conform specificației, împreună cu determinarea atributelor acestora (spațiu ocupat, orientare, componente copil)
* *”Layout Editing & Content Overriding”* (Editor grafic al aspectului si suprascriere de continut)

Componenta realizează corecția unui document convertit la XML printr-o succesiune de interacțiuni cu utilizatorul.

Interacțiunile pot avea urmatoarea natură:

* + - Redimensionări și/sau divizări/uniri de elemente
    - Corecție text

4.3. Restricţiile de implementare

Modulele aplicaţiei trebuie să acopere următoarele restricţii de implementare:

* Portabilitate peste sistemele de operare majore (Windows, Linux, Mac)
* Modului trebuie să poata fi operat (executat) în consistență cu specificația, respectând interfețele asociate
* Datele de intrare și ieșire trebuie să fie conforme cu specificația XSD
* Toate modulele aplicaţiei vor fi dezvoltate utilizând limbajul de programare Java Standard Edition Versiunea 7.

4.4. Interacţiunea dintre componente

Rularea aplicației va aduce utilizatorul în fața unei alegeri privind tipul datelor de intrare:

* Imagine
* XML

În cazul selectării tipului *”imagine”*, se va apela modulul de conversie la XML. În urma, operației se va obține un fișier XML conform specificației XSD, care va centraliza rezultatele detecției de componente și detecției de text.

Dupa acest pas, sau în cazul în care datele sunt prezentate deja sub în format XML, se lanseaza modulul de editare si corecție.

Interfața GUI pune la dispoziția utilizatorului un set de operații conform cu specificația pentru manipularea componentelor documentului, acestea includ:

* Redimensionări
* Unificări
* Diviziuni
* Suprascriere text
* Ștergere entitați
* Adaugare entitați

După această secvența de operații succesive se va exporta rezultatul către un fișier XML conform specificației XSD, constituind rezultatul transformărilor asupra intrării.

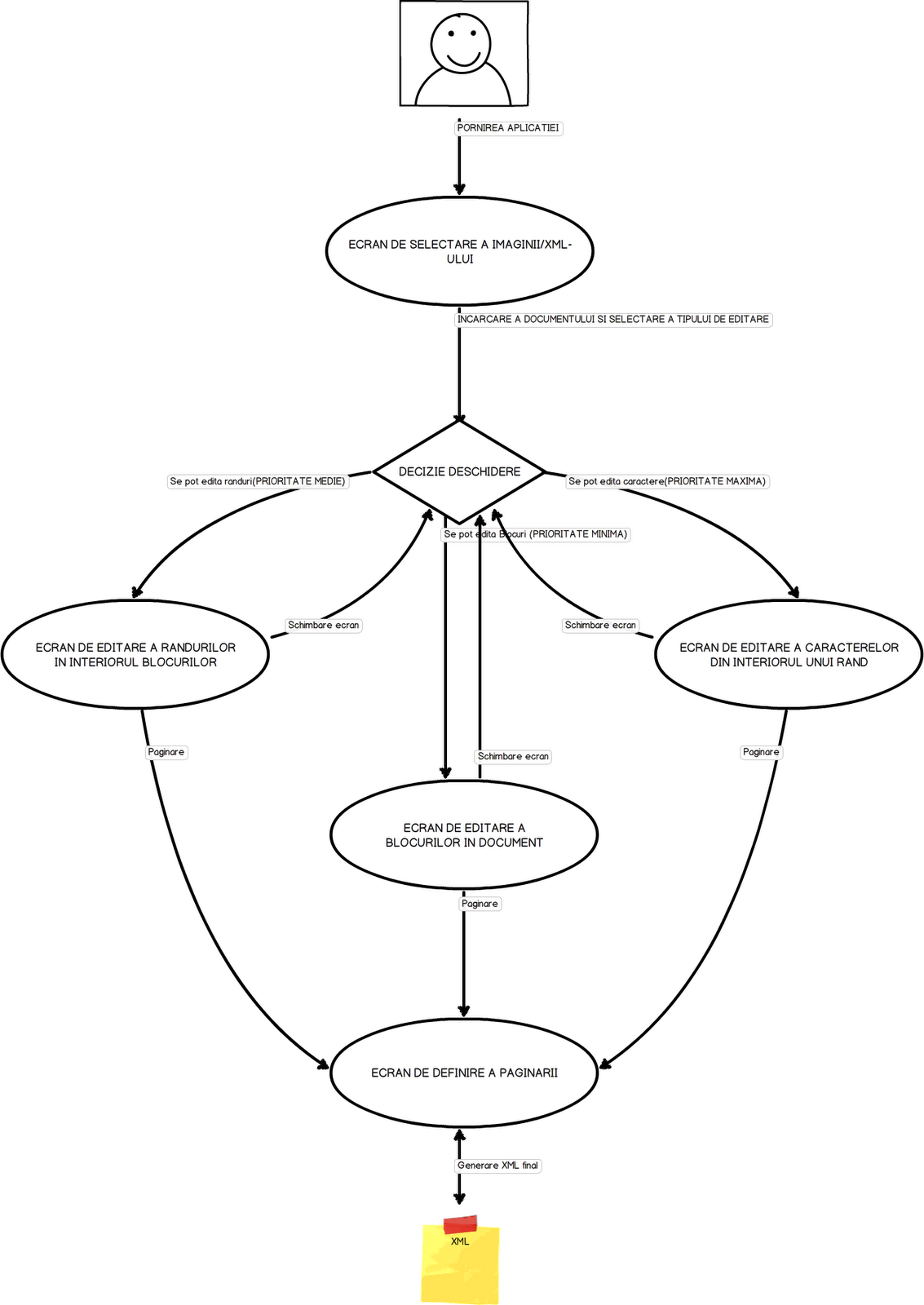
Utilizatorului i se va oferi optiunea de a începe o nouă secvență de operații sau de a părasii aplicația.

5. Modelul interfeței cu utilizatorul

5.1 Succesiunea interfețelor

Modului GUI este reprezentat de o interfață Awt/Swing, respectând workflow-ul de editare și corectare a documentelor XML și de tip imagine stabilit în prezentul proiect.

Succesiunea ecranelor este descrisă de următoarea diagramă:

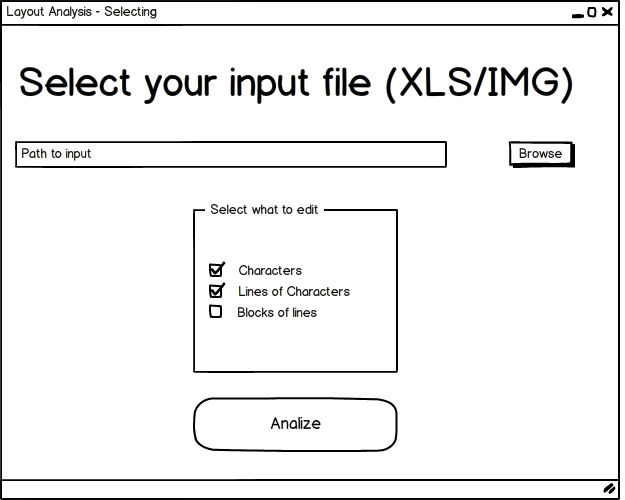


5.2 Ferestrele Aplicație

5.2.1 Fereastra de selectare a documentului

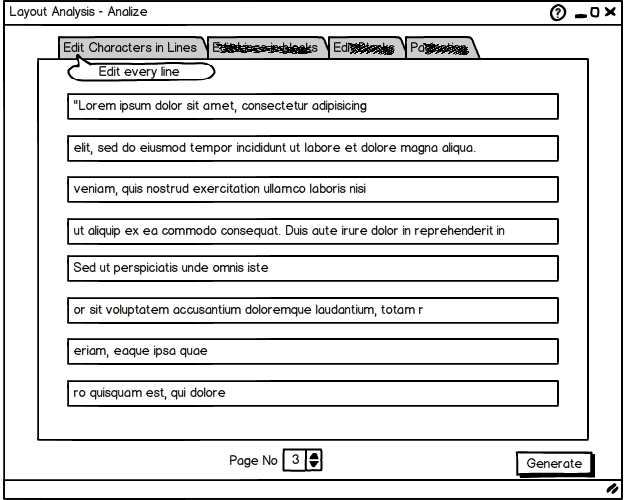
Această fereastră reprezintă dashboard-ul applicatiei. Utilizatorul selectează fișierul pe care dorește să îl editeze / corecteze, alege tipul de editare pe care dorește să îl efectueze și

apasă butonul "Analize" pentru a începe editarea.



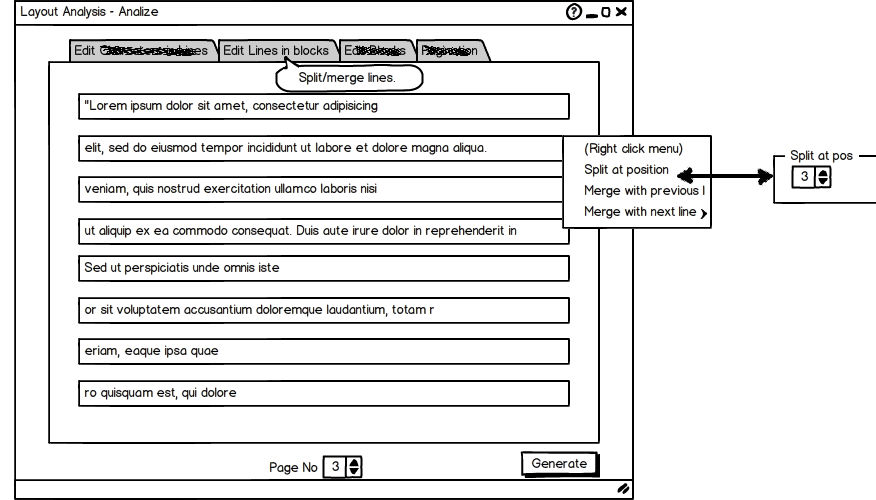
5.2.2 Editarea caracterelor în interiorul liniilor

În funcție de tipul de analiză pe care îl selectează din primul ecran, utilizatorul poate ajunge în ecranul de editare al caracterelor.



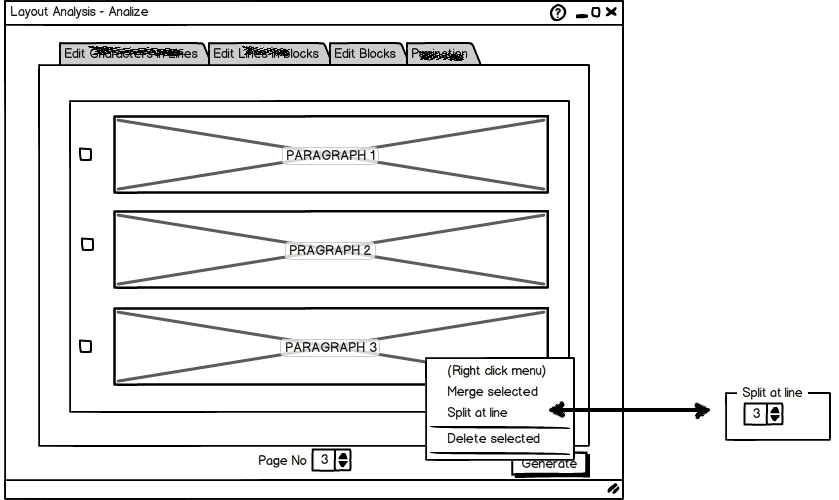
5.2.3 Editarea liniilor în paragrafe

Aici utilizatorul va putea să unească două linii (dacă sunt în același bloc) sau să împartă o linie în două.



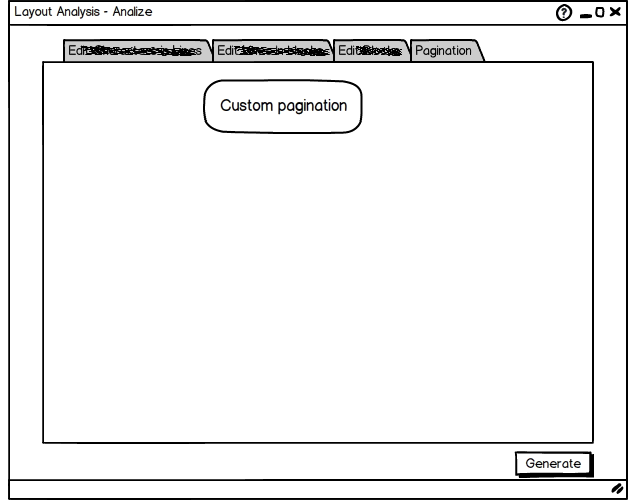
5.2.4 Editarea paragrafelor în document

Utilizatorul va putea să aleagă dintre a concatena sau a împărți paragrafele.



5.2.5 Ecranul de paginare și generare

Din acest ecran utilizatorul poate să suprascrie paginarea detectată de OCR și poate genera fișierul XML final.



6. Elemente de testare

Testarea se va face în mai multe etape, in funcţie de evoluţia proiectului:

1. Testarea elementelor de bază:

- verificarea interfeţei grafice;

- parsarea corectă a XML-ului.

2. Testarea funcţionalitaţii:

- verificare output;

- timpul de procesare trebuie să fie unul mediu (propus de dezvoltatori).

3. Testare finală:

În acest moment proiectul trebuie să fie aproape de finalizare, de aceea vor fi luate in considerare:

- corectitudinea output-ului;

- timpul de răspuns al aplicaţiei (influenţat in mare parte de modul de parsare al XML-ului şi de calcularea poziţiilor blocurilor in imagine);

6.1.Componente critice

Accentul va fi pus pe corectitudinea output-ului, timpul de răspuns in situaţii critice (ex: imagini deformate) şi “corner case”-uri apărute pe parcurul dezvoltării.

Orice element pe care s-a insistat sau de care dezvoltatorul nu este sigur va fi supus verificarii.