

ÉCOLE POLYTECHNIQUE DE L’UNIVERSITE FRANÇOIS RABELAIS DE TOURS

Spécialité Informatique

64 av. Jean Portalis

37200 TOURS, FRANCE

Tél +33 (0)2 47 36 14 31

www polytech univ-tours fr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SPECIFICATION** | | | | |
| **Project :** CDS03 | | Maquette détaillée d’un cahier de spécification | | |
| **Emitter:** | | N. Ragot | | **Owner :** EPU-DI |
| **Date of issue :** | | 28/10/2015 | | |
| **Validation** | | | | |
| Name | | Date | Valid (Y/N) | Comments |
| N. Ragot | | 28/10/2015 | O |  |
|  | |  |  |  |
|  | |  |  |  |
| **History of changes** | | | | |
| Version | Date | Description of the change | | |
| 00 | 11/2008 | Initial version: synthesis of different documents | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |

TABLE OF CONTENTS

Specification booklet

1. Context of the project 5

2. General description 5

3. Description of the external interfaces of the software 6

4. General system architecture 7

5. Description of the features 7

6. Operating conditions 8

Glossary 11

Bibliography 13

Index 15

# spEcifications BOOKLET

This document aims to raise the technical and methodological requirements and for the project. After putting the project into context, it will expose state of the art tools for it, analyse the solution at each stage, define test plans through scenarios and test sets and thus validate the proposal.

The actors of this project are:

* the client, which here are Centre for Advanced Renaissance Studies (fr. Centre d’études supérieures de la Renaissance) (CESR), for which a contact is Rémi Jimenes, lecturer and researcher.
* the Project/Product Owner (fr. Maître d’ouvrage) (fr. MOA), who is Jean-Yves Ramel, professor of computer science, director of Laboratory of Fundamental and Applied Computer Science of Tours (fr. Laboratoire d’Informatique Fondamentale et Appliquée de Tours) (LIFAT) and academic tutor for this project.
* the Project Manager / Scrum Master (fr. Maître d’œuvre) (fr. MOE), Théo Boisseau, an engineering student in his final year of study. He decide on the technical means used to design the product by what was defined by the product owner.

The product owner is responsible for representing the client by ensuring that the deadlines are met and that the product conforms. Thus, he is in charge to review documents such as this one.

## Context of the project

The client expressed the need for easy-to-use interactive software so that its users, historians, could create their own scenarios for extracting elements of content (EOCs) from images of historical documents. These historical documents are mainly Renaissance corpora, accessible from the CESR database, and contain mainly printed or manuscript text, illustrations and page ornaments.

To convert these historical books into accessible digital libraries, LIFAT is developing image processing software that participates in a complete processing chain, including layout analysis, text/illustration separation (i.e. segmentation of content elements), optical character recognition (i.e. OCR) and text transcription. This project focuses on layout analysis and segmentation of content elements of historical documents.

The simplicity of creating extraction scenarios, their reuse and their adaptation to different documents are essential dimensions of the requirement. However, this simplicity should not unduly compromise the reliability and performance of the software. Image processing of historical documents is a particularly difficult task notably because of broken characters, stains, and poor paper quality.

In recent years, the performance of some deep learning techniques has surpassed that of shallow methods established by experts on various image processing tasks. As this progress has made many computer vision tools available, it now seems possible to meet this need with a completely new approach.

### Objectives

This project aims to propose a new approach based on deep learning neural networks to solve this image processing problem.

To this end, the Deep-Agora R&D project aims to build a prototype of an optimisation software capable of extracting textual and decorative elements of content from images of historical documents.

The user should not be responsible for training the models. Therefore, several deep learning models can be created and trained to extract the content elements required in the different use cases of the software.

Due to its nature as a prototype, the system will need to be composed of computational documents combining scripts and good documentation. It must also provide access to training datasets and parameter storage files to reproduce the deep learning models created.

If the objective is achieved, the project can be continued and a scenario creation subsystem can be implemented to deploy the models created within it.

### Hypotheses

*Les hypothèses décrivent tous les facteurs susceptibles de remettre en cause tout ou une partie de la réalisation des spécifications ainsi que d’éventuelles solutions de repli. Par exemple, « Si au cours du projet il se passe X, on fera Y, sinon on fera Z » ou encore « Si on n’arrive pas à faire X, ou si on ne trouve pas la librairie Y, on fera Z ». Il faut bien détailler toutes les alternatives qui se posent en début de projet.*

### Methodological

An Agile project management method will be used to create learning loops to quickly gather and integrate feedback. Therefore, the Scrum method should be preferred in which ideology is to:

* learn from experience
* to self-organise and prioritise
* to reflect on gains and losses to continuously improve

Therefore, contact with the product owner should be maintained as much as possible, as it will help me to improve and learn considerably as the project progresses.

To this end, we set sprints with a fixed duration of 2 weeks. At least one deliverable, containing an e-mail, should be sent to the product owner at least every two weeks and preferably once a week. During the implementation phase, a meeting to get feedback about the product should be scheduled at the end of each sprint.

I use GitHub for configuration management, by creating two different repositories:

* Deep-Agora, which contains the source code of the project
* Deep-Agora\_DOC, which contains all the deliverables of the specification, analysis and modelling part of the first semester

As a project management tool, I will also use GitHub. As of this year, it offers a similar feature to Trello called Projects, an adaptable spreadsheet that can also integrate with my issues and pull requests on GitHub to help me plan and track my work efficiently.

*Cela inclus les références à des documents annexes tels que le plan d’assurance qualité et/ou de test, etc.*

### Description générale

### Project environment

This project is part of a larger research project between CESR and LIFAT. It is currently being carried out as part of a programme for the regional valorisation of old books (mainly dating from the Renaissance), namely the *Humanist Virtual Libraries* controlled by the CESR.

CESR does not have powerful computing machines capable of training deep neural networks, but it has several machines and a large amount of remote and on-premises storage.

Agora, the software developed and published ten years ago by LIFAT to process images of historical documents, is undergoing a complete overhaul in this project. Its technologies need to be updated and, above all, its overhaul should meet the previously unattainable need for simplicity in scenario creation.

Therefore, no takeover of the existing system is planned, as it has to be completely redesigned.

The developer will train deep neural networks, whose task is not intended for the end users of Deep-Agora. This part of the project is to be carried out outside the software system, but within the environment, as an engineer's system.

### User characteristics

End users of Deep-Agora are all historians of CESR.

They have a sufficient but moderate command of computer tools. They often use them but need extensive training or solid documentation to use them in the case of advanced tools with complex functions. They did not have a satisfactory experience with Agora, as its interface was too complex. They do not need user access rights to use Agora.

### System features

*Décrire les fonctions utilisateurs du système et les principaux objets qui constituent le système. Ceci peut être fait au travers d’un ou plusieurs diagrammes de cas d’utilisation généraux par type d’utilisateur.*

Diagram

Description automatically generated

### General structure of the system

*Identifier les principaux composants/éléments du système ainsi que leurs relations. Cela peut être fait au moyen de diagrammes statiques (« objet » au sens large, i.e. classe, composant, déploiement, etc.).*

*Sans être une analyse à part entière, cette partie doit montrer que vous avez une première réflexion sur la structure interne.*

Diagram, schematic

Description automatically generated

## Description of the external interfaces of the software

### Hardware/software interfaces

L’interface matériel/logiciel décrit précisément le matériel informatique et les périphériques, les procédures d’échange d’informations mis en jeu entre eux... On notera donc ici les caractéristiques du matériel qui peuvent avoir une influence sur le logiciel, telles que :

* les normes de communication : protocole d’échange et de raccordement (réseau local ...) ;
* type de liaison (série, parallèle, synchrone, asynchrone, ...) ;
* etc.

### Human/machine interfaces

Il faut spécifier les points suivants :

* ergonomie du système : caractéristiques des messages d’erreur, type de navigation dans le logiciel, etc. ;
* description des formes des éditions sur papier et écrans ;
* mode d’apprentissage de l’interface éventuellement ;
* niveau d’intelligence des interfaces H/M ;
* etc.

Des maquettes ou schémas décrivant ces interfaces ainsi que la charte graphique pourront être présentés ici.

### Software/software interfaces

Il faut spécifier les points suivants :

* moyens d’accès à des systèmes de gestion de base de données, à des bibliothèques logicielles, description de la fréquence des accès, autorisations, etc. ;
* procédures de transferts d’information à distance (échanges d’informations par téléinformatique) ;
* procédures d’échange de messages entre application ;
* etc.

## Spécifications fonctionnelles

Il s’agit de l’expression des besoins fonctionnels. Cette partie a donc comme objectif de décrire l’ensemble des fonctions du système en précisant avec quels composants elles interagissent. Des diagrammes de cas d’utilisation plus détaillés, ainsi que l’arbre hiérarchique des fonctionnalités pourra être fourni ici pour donner une vision plus globale. En outre, chaque fonction sera décrite précisément (cf. ci-dessous). Là encore, il s’agit d’une pré-analyse indispensable à l’évaluation de la complexité de votre projet et à la planification de sa réalisation.

### Definition of the function i

Identification de la fonction i

Présenter la fonction :

* nom de la fonction ;
* rôle, présentation générale ;
* priorité associée à la réalisation de la fonction (primordiale, secondaire, facultative).

Description de la fonction i

Décrire précisément :

* les entrées et les sorties ainsi que les préconditions et postconditions déjà connues, uniquement sous forme textuelle et en langue naturelle (pas dans un pseudo langage algorithmique). Si ces E/S sont connectées à d’autres fonction ou interfaces, le préciser également ;
* les composants avec lesquels cette fonction interagie (données/composants utilisés/modifiés, etc. ;
* le traitement associé à la fonction et à ses interfaces. Il peut s’agir d’une explication ou d’un pseudo-algorithme général précisant les différentes étapes du traitement. Lors de l’analyse, ce dernier pourra être précisément représenté par un diagramme d’activité ; faire référence à un document/article précisant le fonctionnement le cas échéant ;
* si une gestion des erreurs spécifique (hors format des E/S) est prévue et comment celle-ci doit être mise en place si cela est déjà connu (notamment pour les fonctions sensibles).

## Non-functional specifications

### Development constraints and design

Préciser les contraintes liées aux :

* + - * matériels : quelles sont les particularités du matériel qui vont contraindre le développement logiciel;
      * langages de programmation imposés ou adoptés ;
      * logiciels et bibliothèques à utiliser pour le développement;
      * environnements nécessaires : simulateurs, outils logiciels ;
      * bibliothèques de programmes imposées ;
      * protocoles de communication imposés : si nécessaire mettre en annexe une présentation de ces protocoles de communication;
      * etc.

### Functional and operational constraints

Il faut dans ce paragraphe décrire les dispositions qu’il est nécessaire de prendre en compte pour les différentes conditions de fonctionnement su systèmes.

#### Performance

Préciser en termes mesurables, les spécifications temps réel liées à l’utilisation du système :

* du point de vue de l’utilisateur : temps de réponse souhaité, fréquence d’utilisation, temps d’indisponibilité acceptable, etc. ;
* du point de vue de l’environnement : fréquence moyenne d’acquisition d’états ou de mesures, fréquence maximale d’E/S, etc.

#### Capabilities

Décrire les limites des problèmes traitables par le système et les limites des éventuelles extensions comme par exemple :

* nombre max de terminaux ;
* nombre max de points d’acquisition ;
* nombre max de transactions simultanées de tel type, etc. ;
* capacité max de stockage ;
* taille max des données traitées ;
* etc.

#### Operating modes

Décrire les modes d’exploitation du système tels que :

* la mise sous tension ;
* l’arrêt ;
* la reprise de secours ;
* les modes dégradés ;
* etc.

#### Controllability

Il faut décrire, si elles existent, les spécifications particulières permettant de suivre l’exécution d’un traitement (fichier de log, niveaux d’affichages en mode debug, etc.)

#### Security

Indiquer le niveau de confidentialité du système (contrôle d’accès des utilisateurs, mots clefs, mots de passe, etc.). Ceci est directement lié aux différents types d’utilisateurs (cf. 3.2).

#### Integrity

Préciser les protections contre la déconnexion imprévue, les pertes d’information, etc. et quelles sont les procédures à suivre pour restaurer les données du système. Y-a-t-il des situations non protégées ?

### Maintenance and development of the system

Préciser les contraintes liées aux procédures de maintenance :

* curative ou corrective ;
* adaptative ;
* évolutive du système ;
* perfective.

# GLOSSARY

Dans cette partie on doit trouver, classés par ordre alphabétique, les définitions des termes courants utilisés, des termes techniques, abréviation, sigles et symboles employés dans l’ensemble du document.

# BIBLIOGRAPHY

**Aucune source spécifiée dans le document actif.**

Cette dernière partie recense les références techniques sur le projet sur :

* les documents relatifs à l’existant et à l’environnement ;
* les documents sur les méthodes et algorithmes cités ;
* les documents bibliographiques (internes et externes) ;
* les sources d’obtention des documents.

# Index

Cette partie indique les pages où sont traités et mentionnés les sujets et les termes les plus importants du document.

**Aucune entrée d'index n'a été trouvée.**