

Université de Franche-Comté
Faculté des sciences

Master 2 Informatique

Application d'imagerie médicale pour l'aide au traitement du cancer du rein chez l'enfant

THOMAS PRADAL, NIZAR OUARDIANE
ENCADRANTS : JULIEN HENRIET, HERVÉ GUYENNET



Besançon, Le 29 janvier 2016

Table des matières

Remerciements	2
Introduction	3
1 Présentation et notions	4
1.1 Présentation du projet	4
1.2 Format DICOM	4
1.3 Système Multi-Agents	6
1.4 Segmentation	6
1.5 Types de scanner	6
2 Conception de l'architecture	9
2.1 Fonctionnement global	9
2.2 Arbitrage médical	10
2.3 Développement de l'architecture	10
2.4 Utilisation des agents	15
3 Recherche sur l'arbitrage médical	16
3.1 Définition et objectif	16
3.2 Premières réflexions	16
3.3 Redéfinition des exigences	17
3.4 Développement	18
4 Automatisation des tâches	21
4.1 Raisonnement à Partir de Cas	21
4.2 Évaluation des images	22
4.3 Utilisation dans le cadre de la segmentation	23
4.4 Amélioration de l'architecture	24
4.5 Architecture finale	25
5 Conclusion	30
5.1 Bilan personnel	30
5.2 Perspectives	30
Table des figures	31
Bibliographie	32
Annexes	33
5.3 Diagrammes de classe	33

Remerciements

Nous tenons à remercier dans un premier temps MM. Julien Henriet et Hervé Guyennet pour nous avoir proposé un sujet de projet atypique et intéressant ainsi que pour nous avoir suivi et consacré du temps tout au long de sa réalisation.

Nous remercions également Yann Chaussy pour le partage de ses connaissances.

Introduction

Aujourd’hui, afin de diagnostiquer et de traiter le cancer du rein chez l’enfant et le nouveau-né, les médecins s’appuient sur les images obtenues par scanner. Ces images sont ensuite traitées par des algorithmes et programmes informatiques de segmentation permettant de distinguer les différents organes et les tumeurs cancéreuses. Ces segmentations permettent ensuite d’obtenir une représentation virtuelle en 3D de la tumeur, du rein, et des organes avoisinants.

Malheureusement, ces programmes de segmentation (basés sur les intensités lumineuses obtenues lors du scanner), ne sont pas complètement affinés et distinguent mal des organes aux intensités lumineuses proches. Les radiologues ont donc besoin d’intervenir et de délimiter les différents organes sur plusieurs scans. Ces opérations prennent du temps et c’est un temps précieux que perdent les différents intervenants médicaux dans la réalisation du processus.

Actuellement les chirurgiens pédiatres sont obligés de passer par des entreprises spécialisées dans le traitement d’images afin d’obtenir la représentation virtuelle en 3D de l’ensemble des organes et de la tumeur. Cette prestation extrêmement onéreuse ne satisfait pas tous leurs besoins.

Le projet d’application d’imagerie médicale pour l’aide au traitement du cancer du rein chez l’enfant intervient alors et propose un outil informatique basé sur des techniques d’Intelligence Artificielle pour imiter le processus mis en œuvre par les radiologues. Le but étant de les décharger de cette tâche chronophage ainsi que de leur offrir des services supplémentaires qui les aideront dans les processus pré-opératoires et opératoires.

Chapitre 1

Présentation et notions

1.1 Présentation du projet

Le but du projet est de se familiariser avec les images au format DICOM, format d'image utilisé dans le milieu médical. L'objectif principal ensuite est de créer une architecture distribuée du système à base d'agents qui permettrait de réaliser les prestations attendues par les médecins. En parallèle, un projet sur la réalisation du processus de segmentation est réalisé par un autre binôme.

Ce projet s'inscrit dans le cadre d'un projet de plus grande envergure : le projet SAIAD (Segmentation Automatique de reins tumoraux chez l'enfant par Intelligence Artificielle Distribuée). Il s'agit d'un projet mené conjointement par l'Université de Franche-Comté, le Centre Hospitalier Régional Universitaire, Covalia, et l'École Polytechnique Fédérale de Lausanne.

Les tumeurs du rein sont les tumeurs abdominales les plus fréquentes chez l'enfant, et sont diagnostiquées entre 30 et 47 mois. Actuellement, ce diagnostic se base sur l'échographie pour détecter la tumeur et le scanner pour la localiser précisément.

Or il faut analyser en moyenne 250 images par patient.

Le projet SAIAD a donc pour objectifs de :

- construire une base de connaissances permettant d'automatiser la segmentation,
- utiliser l'intelligence artificielle pour optimiser la segmentation,
- permettre la représentation en 3 dimensions des viscères segmentés.

Cette représentation en 3 dimensions peut alors servir de base pour les diagnostics afin de les simplifier par rapport à une série de scanners.

Or la transformation des scanners en images en 3 dimensions de façon automatique est actuellement impossible car il n'existe pas de logiciel capable de segmenter une image sans intervention humaine. La segmentation des images devra donc permettre de détecter tous les éléments présents dans l'abdomen sur des coupes à des niveaux différents sans interaction avec l'utilisateur.

1.2 Format DICOM

DICOM [1] (en anglais : *Digital Imaging and Communications in Medicine*), est une norme standard pour la gestion informatique des données issues de l'imagerie médicale.

Le format a été créé en 1985 par l'ACR (American College of Radiology) et la NEMA (National Electric Manufacturers Association) dans le but de standardiser les données transmises entre les différents appareils de radiologie. Ce standard définit un format de fichier mais aussi un protocole de transmission des données (basé sur TCP/IP).

L'objectif du standard DICOM est de faciliter les transferts d'images entre les machines de différents constructeurs. En effet, avant la généralisation de ce format, chaque constructeur de matériel d'imagerie utilisait un format de données propriétaire, entraînant d'importants problèmes de gestion et de maintenance (incompatibilités, coût, perte d'information) dans les établissements de santé.

Le tirage des clichés sur papier argentique n'est plus incontournable, ce qui diminue de beaucoup le coût d'une radiographie.

Le suivi médical des patients, surtout en cas de pathologie lourde nécessitant souvent le transfert d'un établissement de santé à un autre en fonction des moyens et compétences disponibles, a directement bénéficié de l'instauration de cette norme. Les images au format DICOM accompagnant les dossiers médicaux sont lisibles sur tout matériel informatique compatible, et rendent obsolète le transport des clichés par les moyens de communication traditionnels, principalement les envois par courrier.

Le format DICOM est donc constitué de nombreux champs correspondants aux différentes informations du patient ainsi que la matrice de pixel stockant l'image.



FIGURE 1.1 – Exemple d'image DICOM anonymée

1.3 Système Multi-Agents

En informatique, un agent est un logiciel qui agit de façon autonome. C'est un programme ou processus qui accomplit des tâches à la manière d'un automate et en fonction de ce que lui a demandé son auteur.

Un système Multi-Agents [2] (SMA) est alors un système composé d'un ensemble d'agents, situés dans un certain environnement et interagissant selon certaines relations.

Ainsi, pour le projet SAIAD, les agents seront utilisés dans une optique de création d'une Intelligence Artificielle autonome qui permettra d'appliquer une segmentation efficace pour toute image de scanner qui lui sera fournie.

1.4 Segmentation

La segmentation d'image est une opération de traitement d'images qui a pour but de rassembler des pixels entre eux suivant des critères prédéfinis. Les pixels sont ainsi regroupés en régions, qui constituent un pavage ou une partition de l'image. Il peut s'agir par exemple de séparer les objets du fond.

Si l'homme sait naturellement séparer des objets dans une image c'est grâce à des connaissances de haut niveau (compréhension des objets et de la scène). Mettre au point des algorithmes de segmentation de haut niveau (chaque région est un objet sémantique) est encore un des thèmes de recherche les plus courants en traitement d'images.

La segmentation est une étape primordiale en traitement d'image. À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes :

- La segmentation fondée sur les régions (en anglais : region-based segmentation). On y trouve par exemple : la croissance de région (en anglais : region-growing), décomposition/fusion (en anglais : split and merge).
- La segmentation fondée sur les contours (en anglais : edge-based segmentation).
- La segmentation fondée sur la classification ou le seuillage des pixels en fonction de leur intensité (en anglais : classification ou thresholding).
- La segmentation fondée sur la coopération entre les trois premières segmentations.

Dans notre cas, la segmentation va permettre la mise en évidence des organes sur chaque image. Il pourra alors être possible de distinguer la tumeur du rein, ainsi que visualiser les différentes cavités et leur emplacement.

1.5 Types de scanner

Le scanner utilise les mêmes rayons X, plus ou moins absorbés selon la densité des tissus traversés, que la radiographie classique. Les rayons X ne sont pas reçus sur un film, mais sur une plaque radiosensible reliée à un ordinateur.

Plusieurs millions de mesures sont effectuées en quelques minutes. L'ordinateur les utilise pour reconstituer l'image des organes abdominaux plan par plan, par coupes séparées de quelques millimètres.

Il existe deux types d'utilisations du scanner permettant d'obtenir des images différentes par leur contenu et les informations qu'elles renseignent.

1.5.1 Scanner sans injection

Le scanner sans injection correspond à l'acquisition des images avant l'injection intraveineuse de produit de contraste. Il s'agit du premier temps d'acquisition des images. Il peut être utile et suffisant dans certaines situations (recherche de lithiase urinaire, recherche d'un hématome cérébral...) mais ne permet pas d'obtenir des informations suffisantes dans notre cas.



FIGURE 1.2 – Exemple d'image scanner sans injection

1.5.2 Avec injection

Le scanner avec injection correspond à l'acquisition des images après l'injection intraveineuse de produit de contraste. Une injection de produit de contraste iodé par voie intraveineuse permet de rehausser les contrastes entre les différentes structures et permet ainsi une meilleure analyse de l'image.

Différents temps d'acquisition sont alors possibles en fonction du délai entre l'injection du produit de contraste et l'acquisition des images. Dans notre cas, les temps tardifs et vasculaires seront utilisés. Ce sont les deux temps qui permettront d'obtenir le plus d'informations utiles.

Temps vasculaire

L'acquisition au temps vasculaire se fait quelques secondes à quelques minutes après l'injection, le produit de contraste va se situer dans le réseau vasculaire du patient.

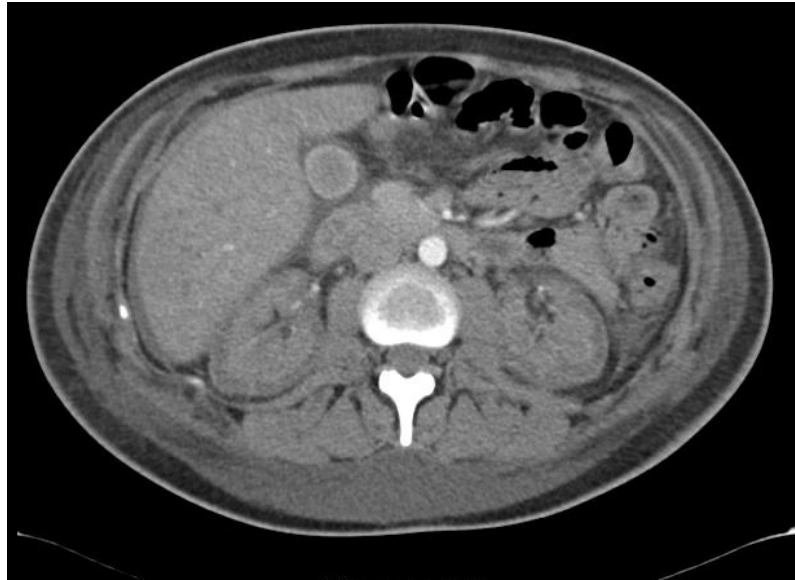


FIGURE 1.3 – Exemple d'image scanner avec injection au temps vasculaire

Temps tardif

L'acquisition au temps tardif se fait environ 10 minutes après l'injection du produit iodé. Le produit de contraste va être éliminé par voie urinaire. On va ainsi obtenir une bonne opacification de l'ensemble des voies urinaires (cavités urinaires intrarénales, pyélon, uretère, vessie) et obtenir des informations supplémentaires.

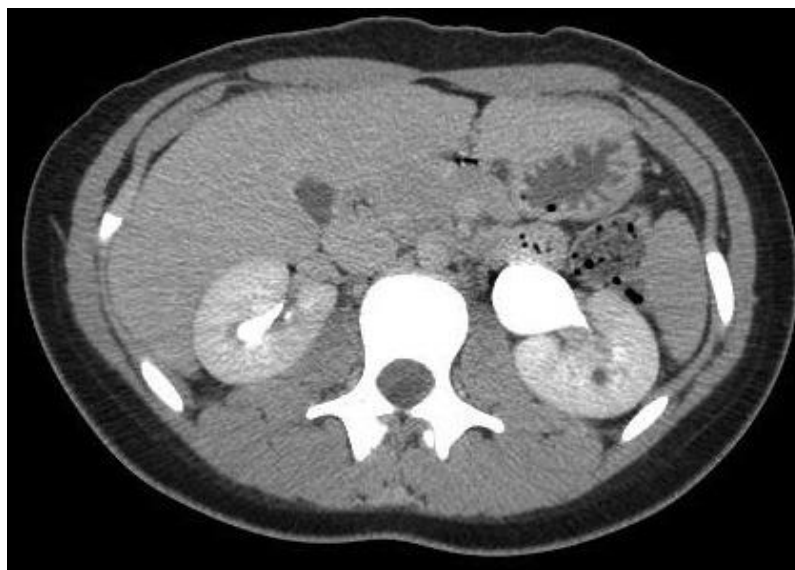


FIGURE 1.4 – Exemple d'image scanner avec injection au temps tardif

Chapitre 2

Conception de l'architecture

Le but de notre projet étant d'établir une base de travail pour le projet SAIAD, il a fallu penser une architecture la plus optimisée possible tout en permettant d'accomplir efficacement les tâches demandées et d'obtenir le résultat attendu.

2.1 Fonctionnement global

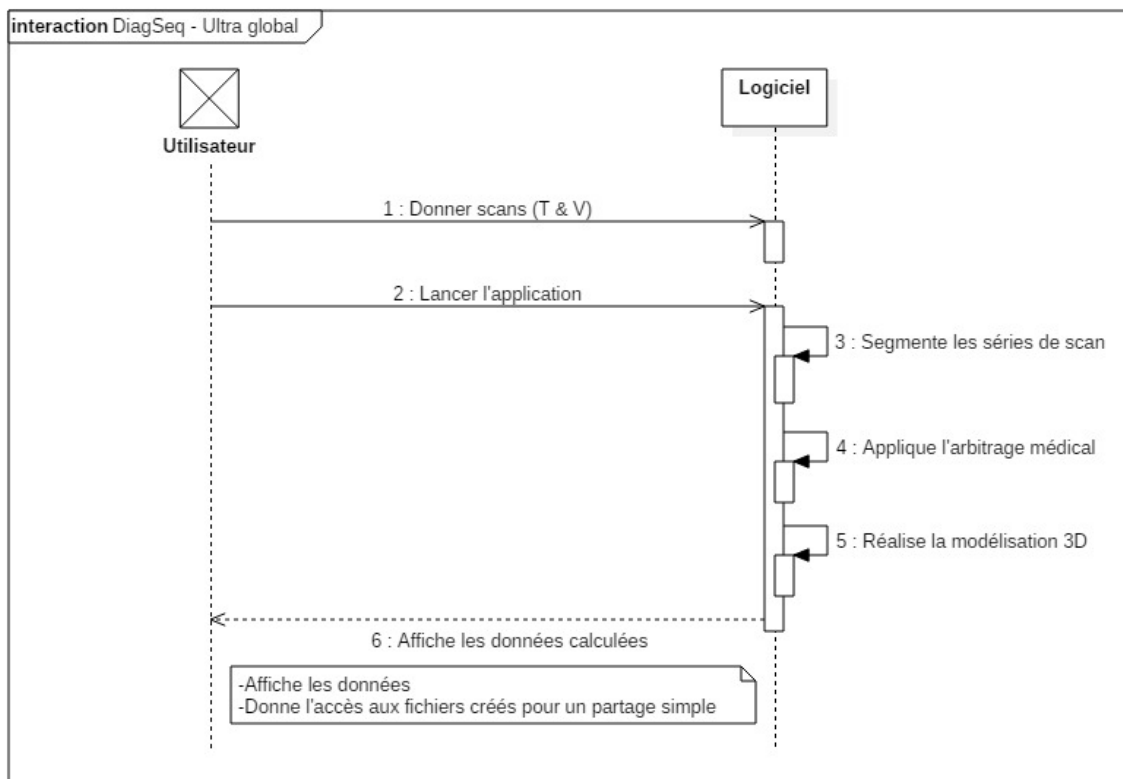


FIGURE 2.1 – Diagramme de séquence - Architecture "ultra" globale du système

La première architecture, proposée sous forme de diagramme UML de séquence (Figure 5.3), permet avant tout de définir et présenter les principaux acteurs du système que sont l'utilisateur et le logiciel ainsi que les interactions entre ces derniers.

Le logiciel s'utilisera donc après l'obtention des séries de scanner du patient. L'utilisateur les fournissant au logiciel, ce dernier pourra alors effectuer les tâches de segmentations, d'arbitrage médical et de modélisation 3D.

L'ensemble des résultats pourront être consultés et récupérés par les médecins afin de les étudier, stocker, partager facilement.

Cette première architecture nous permet ensuite d'introduire le concept d'arbitrage médical.

2.2 Arbitrage médical

L'arbitrage médical est une fonctionnalité ayant pour origine un besoin de rassemblement des informations des images segmentées. En effet, chaque type d'acquisition du scanner utilisé ici permet l'obtention d'informations différentes. Cette fonctionnalité aura alors pour but de rassembler ces informations dans une seule et même image.

Le concept étant décrit dans le chapitre 3 - *Recherche sur l'arbitrage médical*, il ne sera pas plus expliqué dans ce chapitre.

2.3 Développement de l'architecture

L'architecture de base étant créée, elle peut maintenant être développée tout en restant simple avec une segmentation fixe sans Intelligence Artificielle permettant d'automatiser sa sélection.

On peut donc distinguer les différents agents qui agiront lors des tâches à accomplir par le logiciel.

2.3.1 Partie segmentation

L'utilisateur ayant fourni au logiciel les séries de DICOM générées par le scanner en temps tardif et vasculaire, la tâche de segmentation peut être lancée. Elle a alors pour but de segmenter l'ensemble des images desdits DICOM.

L'objectif pour le logiciel va être, pour chaque image, d'appliquer les algorithmes nécessaires à la segmentation. Ces algorithmes pouvant être longs, il apparaît intéressant d'effectuer cette tâche parallèlement afin de segmenter plusieurs images en même temps et ainsi accélérer l'obtention des résultats.

Dans notre cas, la segmentation des images en temps tardif et vasculaire se fera en parallèle pour une même coupe. On peut malgré tout imaginer cette tâche s'effectuer séquentiellement sur une machine qui n'aurait pas les moyens de faire autrement.

Les coupes seront segmentées une à une de manière séquentielle. Également, on peut imaginer avoir les moyens de paralléliser cette action et ainsi segmenter plusieurs coupes en même temps et accélérer le processus.

2.3.2 Redéfinition des agents

Suite à la définition du fonctionnement de chacune des tâches citées précédemment, des agents font leur apparition et se voient déléguer les tâches par l'agent logiciel.

Segmentation sur coupe

Le système segmentant les images en temps tardif et vasculaire d'une même coupe à la fois, un agent *Segmentation sur coupe* permet de faire le lien entre le logiciel lui fournissant les deux dites images et l'application même de la segmentation.

Il va alors être capable de paralléliser la segmentation sur les deux images de sa coupe, récupérer le résultat et le retourner à l'agent logiciel. Il lancera également le processus d'arbitrage médical, en récupérera le résultat et les retournera à l'agent logiciel.

Segmentation

L'agent de *Segmentation* est l'exécuteur même des algorithmes de segmentation, il récupère une image en entrée, applique l'algorithme et renvoie le résultat. Dans l'optique d'une évolution vers l'automatisation de la tâche de segmentation, cet agent pourra appliquer les différents types d'algorithmes de segmentations connus avec les paramètres donnés.

Il est exécuté par l'agent de *Segmentation sur coupe*.

Arbitrage médical

L'agent *Arbitrage médical* intervient logiquement pour appliquer ledit arbitrage entre les images en temps tardif et vasculaire d'une même coupe et retourne le résultat.

Il est exécuté par l'agent de *Segmentation sur coupe* à partir des images segmentées préalablement.

Logiciel

Dans cette première évolution de l'architecture, l'agent logiciel récupérera les images de la même façon que dans la version précédente. L'utilisateur lui fournit les séries de DICOM générés par le scanner.

Ayant maintenant l'agent *Segmentation sur coupe* à disposition, le logiciel délègue la tâche de segmentation coupe par coupe en lui fournissant à chaque fois les images en temps tardif et vasculaire.

Il récupère l'ensemble des résultats (images segmentées et arbitrées) pour pouvoir ensuite les mettre à la disposition de l'utilisateur.

2.3.3 Nouvelle architecture

Dans les images suivantes (Fig 2.2 et 2.3), l'architecture de l'application est décrite à l'aide de diagrammes UML de séquence et d'activité.

On y retrouve les principes décrits précédemment.

1. L'utilisateur fournit les séries de DICOM en temps tardif et vasculaire au logiciel
2. L'utilisateur lance l'application
3. Le logiciel lance le processus de segmentation des images sur la première coupe
4. Le logiciel lance l'agent de segmentation par coupe en lui fournissant la coupe z de DICOM
5. L'agent de segmentation par coupe lance la segmentation sur les DICOM de la coupe z
6. La segmentation des deux DICOM de la même coupe est parallélisée
7. L'agent de segmentation par coupe récupère les DICOM segmentés
8. L'agent de segmentation par coupe lance l'arbitrage médical sur les DICOM segmentés
9. L'arbitrage médical est appliqué et le résultat retourné
10. L'agent de segmentation par coupe récupère l'image arbitrée et la retourne au logiciel avec les DICOM segmentés
11. Si l'ensemble des coupes n'a pas été traité, on retourne à l'étape 4 avec le z suivant
12. Sinon le logiciel a terminé son activité et met à disposition l'ensemble des données à l'utilisateur.

Le processus de modélisation 3D ne faisant pas partie des objectifs à mettre en place dans l'architecture durant ce projet, il n'apparaît pas dans les diagrammes.

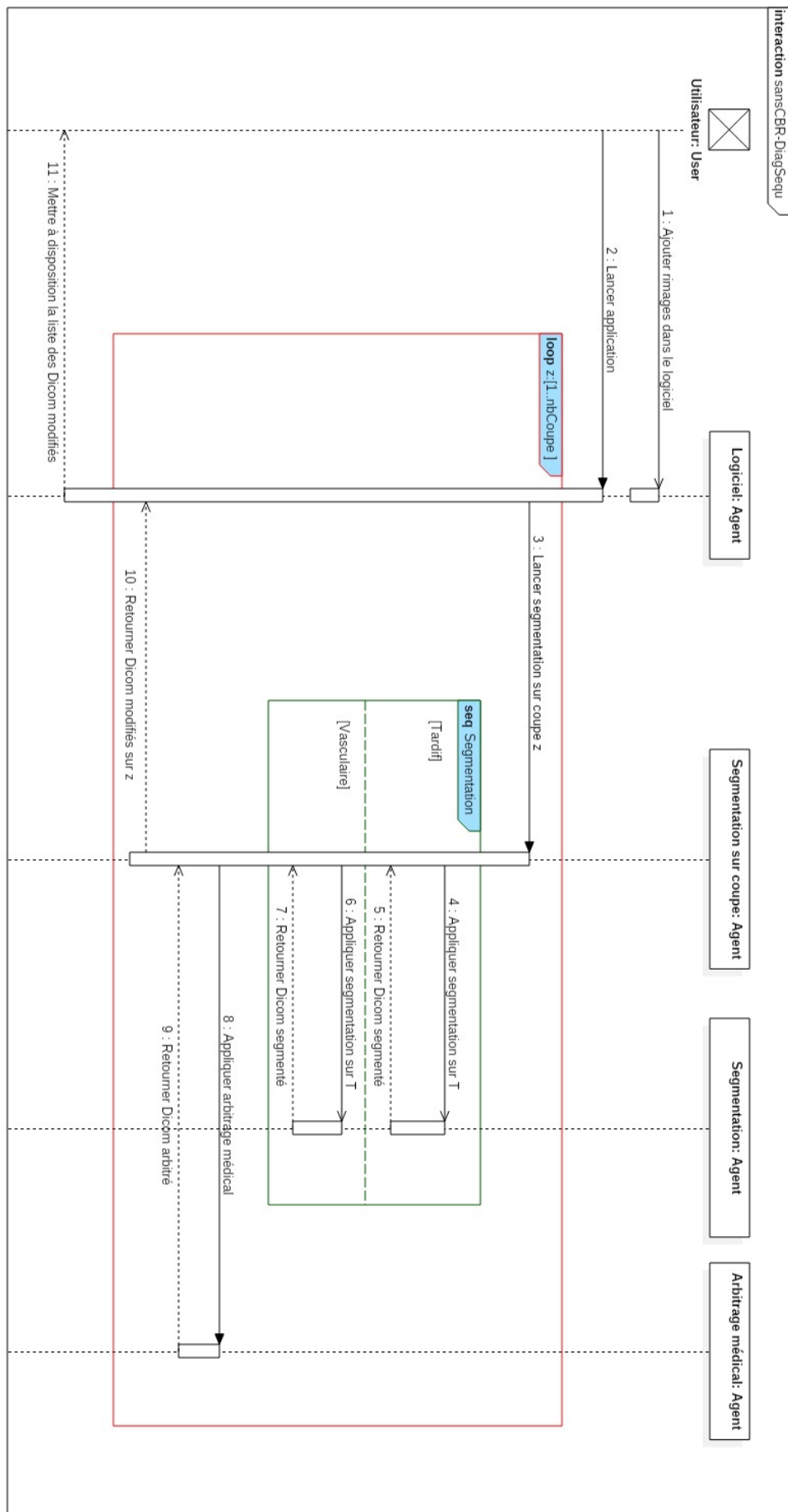


FIGURE 2.2 – Diagramme de séquence - Architecture du système

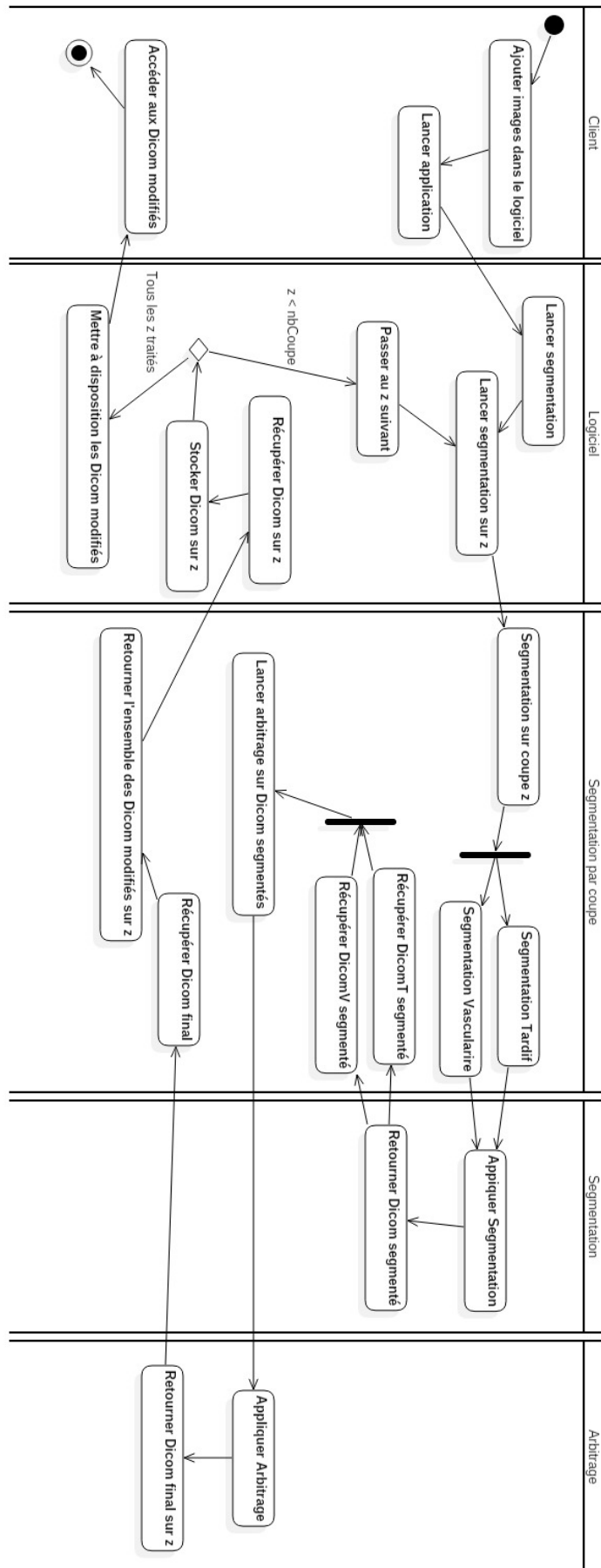


FIGURE 2.3 – Diagramme d'activité - Architecture du système

2.4 Utilisation des agents

Rappelons qu'un agent est un processus capable de réfléchir par lui-même, il est autonome et peut donc prendre des décisions de son propre chef.

Dans notre cas, l'ensemble des exécuteurs de tâche sont définis en tant qu'agents. Si nous nous mettons dans le cas d'une exécution complètement séquentielle de l'application, chacun des agents de l'architecture (voir Figure 2.2) ne correspond plus qu'à de simples fonctions appelées les unes par les autres.

La distinction de chacun des acteurs de l'architecture en tant qu'agent devient intéressante dans le cadre de la parallélisation des tâches. En effet, si un acteur comme la *Segmentation sur coupe* devient capable de prendre la décision de paralléliser certaines tâches dès qu'il en a les moyens, alors la notion d'agent prend tout son sens. Dès lors, tous les acteurs peuvent, en tant qu'agent, décider de paralléliser leurs tâches quand ils le peuvent et ainsi gagner du temps à chaque exécution.

Chapitre 3

Recherche sur l'arbitrage médical

3.1 Définition et objectif

L'arbitrage médical est une fonctionnalité de traitement sur image segmentée permettant un gain d'informations à partir des images au temps tardif et vasculaire.

Cette fonctionnalité absente des services payants actuellement utilisés par les médecins leur permettra de mieux visualiser l'ensemble des différentes cavités du rein et leurs emplacements par rapport à l'organe et sa tumeur.

En effet, le temps vasculaire permet de bien identifier le réseau vasculaire (artères et veines) et la plupart des organes (cortex rénal, rate, foie...). Cependant, il est parfois difficile de différencier la tumeur des cavités urinaires intra-rénales sur ce temps là. Cette information est cependant primordiale car elle permet de savoir si la tumeur envahit les cavités urinaires.

A l'inverse, le temps tardif permet de bien repérer l'ensemble de la voie urinaire, mais, le produit de contraste ayant quitté le système vasculaire, il devient difficile de différencier tumeur et organe (cortex rénal, foie, rate...).

L'idée est donc de pouvoir apporter un maximum d'informations sur une même image en utilisant les avantages de chacune des acquisitions.

3.2 Premières réflexions

La définition finale du processus d'arbitrage médical n'a été défini que plus tard dans le projet. L'idée de départ partait alors d'un principe de fusion entre les images segmentées en temps tardif et vasculaire.

Sur cette base, il a été imaginé un système permettant la fusion des deux images en évaluant les pixels de chaque image un par un et de garder l'optimal pour l'image finale.

Dans l'attente de plus d'informations de la part des médecins concernant les critères d'évaluation d'un pixel, les systèmes à base de règles ont été étudiés pour être capable d'appliquer lesdits critères.

Système à base de règles

D'une manière générale, un système expert est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier. Il s'agit de l'une des voies tentant d'aboutir à l'intelligence artificielle.

Plus précisément, un système expert est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connus. Il peut servir notamment comme outil d'aide à la décision.

À l'aide de ce système couplé aux règles connues données par les médecins, l'arbitrage médical aurait été capable d'appliquer le processus de fusion entre les deux images segmentées et d'obtenir le résultat attendu.

3.3 Redéfinition des exigences

Finalement, après concertation avec les médecins, la définition de l'arbitrage médical a été précisée. Il ne s'agit donc plus d'évaluer les images segmentées pixel par pixel mais, toujours sur une même coupe, de récupérer les zones rénales mises en évidence par l'image segmentée en temps tardif et de les intégrer à l'image segmentée en temps vasculaire en leur appliquant une légère transparence.

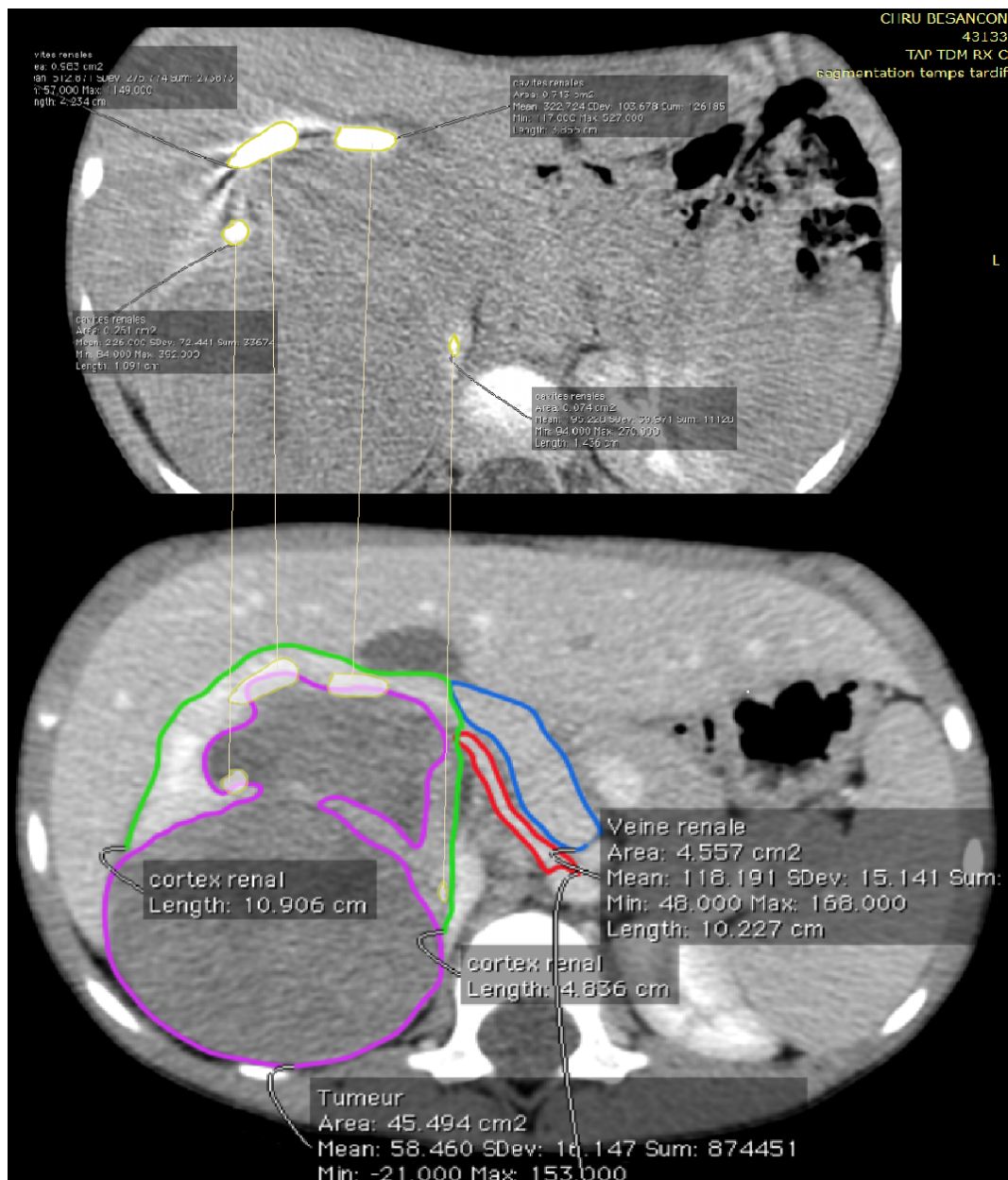


FIGURE 3.1 – Exemple d'arbitrage médical

L'illustration de ce principe est représenté dans la figure 3.1.

Les deux images sont respectivement les images en temps tardif et vasculaire.

La segmentation est une segmentation appliquée à la main et n'est donc pas représentative d'une segmentation effectuée par algorithmes.

De la même façon, l'arbitrage médical effectué ici est manuel et ne sert que d'illustration à la fonctionnalité.

Les zones segmentées sont par ailleurs les zones d'intérêt qui seront utiles aux médecins pédiatres.

Enfin, de cette redéfinition naissent de nouvelles problématiques de recalibrage ainsi qu'un besoin de repenser les technologies à utiliser pour appliquer l'arbitrage médical.

3.4 Développement

3.4.1 Problématique du recalibrage

La problématique du recalibrage existe du fait que les séries de scanner en temps tardif et vasculaire sont effectuées avec plusieurs minutes d'intervalle. De ce fait, la position du patient comme celle des tissus mous/organes n'est pas toujours la même sur les deux séries d'images.

Le principe de l'arbitrage médical étant de copier des parties de l'image en temps tardif dans l'image en temps vasculaire, il apparaît nécessaire que la partie copiée se retrouve exactement au même endroit par rapport au corps du patient et non aux coordonnées de cette zone dans son image d'origine.

Afin de résoudre ce problème, le choix d'un point fixe dans le corps du patient permet de situer n'importe quelle zone par rapport à ce point et ainsi éviter un grave problème de non-exactitude de la position de la zone copiée.

Choix du point de repérage

Il est difficile de trouver un point fixe dans la zone de la partie abdominale ciblée pour ce type de scanner. Ce point de repérage doit également être commun à un maximum de patient afin de rendre le processus d'arbitrage totalement efficace.

Les tissus mous pouvant subir des déplacements, à cause notamment de la respiration, les vertèbres de la colonne vertébrale sont la meilleure option.

La vertèbre a donc pu être défini avec l'aide des médecins pédiatres. L'option la plus optimale se trouvant être la vertèbre lombaire L2 pour sa position par rapport aux reins.

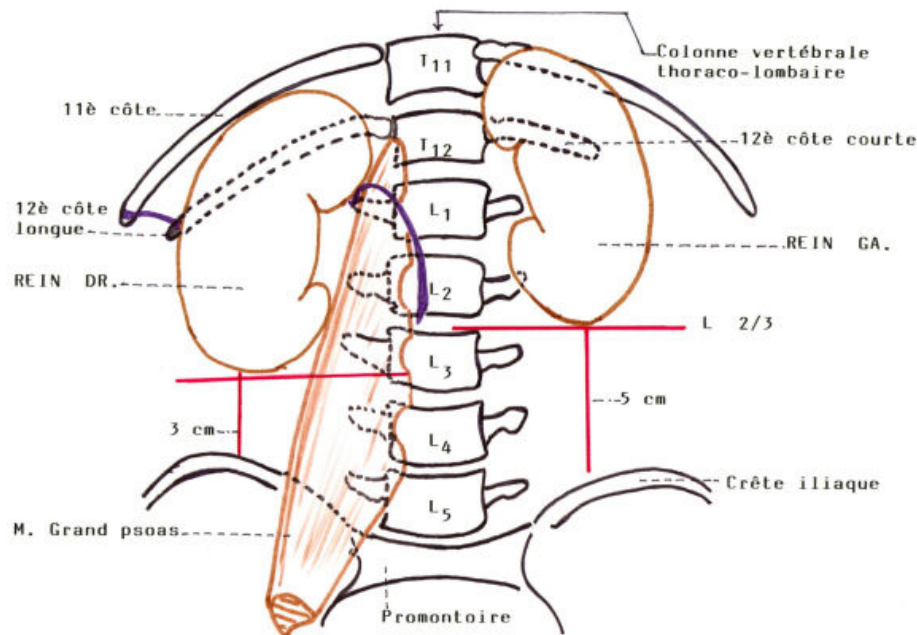


FIGURE 3.2 – Emplacement des reins par rapport à la vertèbre L2

3.4.2 Définition des besoins

Format d'image

Les différentes bibliothèques de traitement d'images n'étant pas compatibles avec les images au format DICOM, la première étape dans la réalisation de l'arbitrage médical est de convertir ladite image en un format utilisable.

Plusieurs projets open source de manipulation des fichiers au format DICOM existent et pourront être utilisés dans cette optique. Dans une logique de cohésion avec le groupe travaillant sur le projet de Segmentation des images médicales, la bibliothèque Imebra [4] pourra être utilisée afin de convertir les images DICOM aux formats PNG ou JPEG.

Dans l'architecture proposée, le processus de segmentation précédant l'arbitrage médical nécessite déjà une conversion de l'image DICOM vers un format PNG ou JPEG. On peut donc imaginer que l'arbitrage peut se passer de cette conversion.

Cependant, dans l'optique de rendre le module d'arbitrage médical totalement autonome, lui permettre de convertir les images au format DICOM pourrait donner la possibilité de (re)lancer l'arbitrage médical sur une série d'images DICOM déjà segmentées. L'arbitrage médical serait alors indépendant et utilisable sur des séries d'images déjà segmentées et non-arbitrées.

Il sera par ailleurs préférable de choisir une conversion au format PNG pour éviter les pertes de données dues à la compression du format JPEG.

Régions d'intérêt

Une fois les images segmentées converties et fournies au module d'arbitrage médical, ce dernier peut réaliser sa tâche de copie des régions d'intérêt d'une image à l'autre. Une région d'intérêt (en anglais ROI pour Region Of Interest) étant une zone intéressante et mise en évidence par le processus de segmentation.

Pour ce faire, une librairie de traitement d'images est nécessaire. La librairie OpenCV [5] (pour Open Computer Vision) est une bibliothèque graphique libre permettant notamment de nombreux traitements d'images et notamment la détection de zones par la couleur ainsi que de transposition d'une zone d'une image à une autre.

Ainsi, pour chaque zone d'intérêt mise en évidence par la segmentation, l'arbitrage médical la détectera, la stockera lui appliquera une légère transparence pour ne pas effacer l'information dans l'image en temps vasculaire et la lui copiera en utilisant le recalibrage si nécessaire.

Déroulement du processus

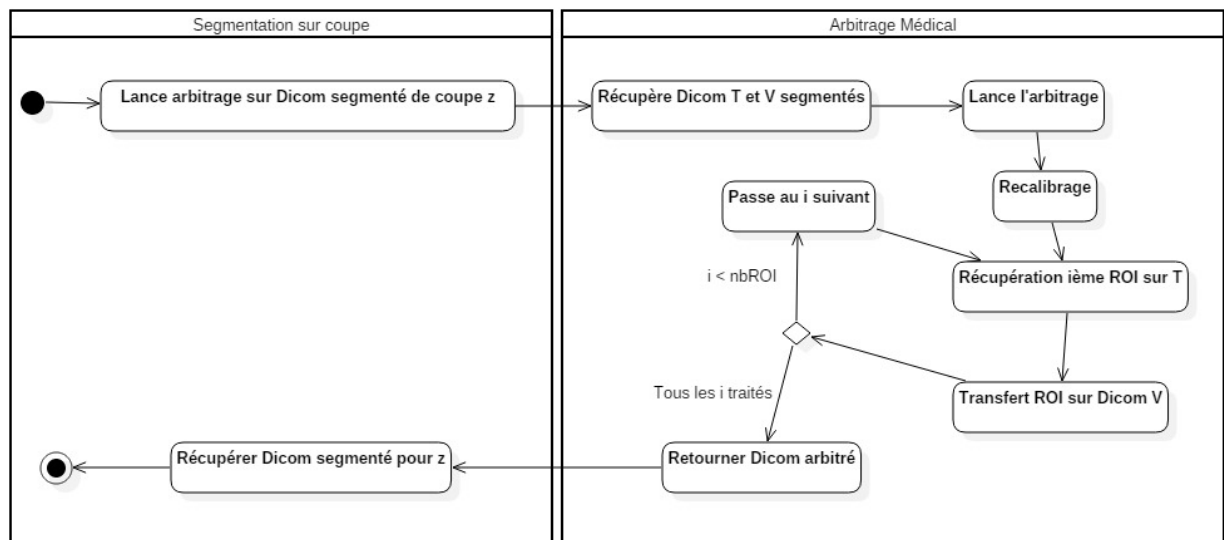


FIGURE 3.3 – Diagramme d'activité - Arbitrage médical

Chapitre 4

Automatisation des tâches

Le but final de l'application étant de rendre le processus complètement automatique, il est nécessaire de faire évoluer l'architecture actuelle ne permettant qu'une segmentation dont le type et les paramètres doivent être modifiés par l'utilisateur.

Dans cette optique il va être nécessaire de déployer un agent dans l'architecture, capable d'évaluer une image et de déterminer le type de la segmentation à appliquer ainsi que ses paramètres, afin d'obtenir des résultats optimaux après le processus de segmentation.

Pour ce faire, il existe une solution permettant à un agent de se servir d'une base de faits pour assimiler le problème à un problème déjà rencontré auparavant et ainsi le résoudre problème.

4.1 Raisonnement à Partir de Cas

Pour résoudre les problèmes de la vie quotidienne, l'homme fait naturellement appel à son expérience. Il se remémore les situations semblables déjà rencontrées. Puis il les compare à la situation actuelle pour construire une nouvelle solution qui, à son tour, s'ajoutera à son expérience.

Le raisonnement à partir de cas (RàPC) [3] (en anglais Case Based Reasoning (CBR)) copie ce comportement humain. Il résout les problèmes en retrouvant des cas analogues dans sa base de connaissances et en les adaptant au cas considéré. On utilise indirectement les données pour retrouver les cas proches, à partir desquels on va générer une solution.

Un système RàPC dispose donc d'une base de cas. Chaque cas possède une description et une solution. Pour utiliser ces informations, un moteur est aussi présent. Celui-ci va retrouver les cas similaires au problème posé. Après analyse, le moteur fournit une solution adaptée qui doit être validée. Enfin le moteur ajoute le problème et sa solution dans la base de cas.

Ces étapes sont illustrées dans la figure 4.1.

On y distingue la base de cas au centre de l'application ainsi que le processus de recherche de cas similaires au problème à traiter, la solution retrouvée grâce à ces cas similaires et l'apprentissage par cette solution au problème.

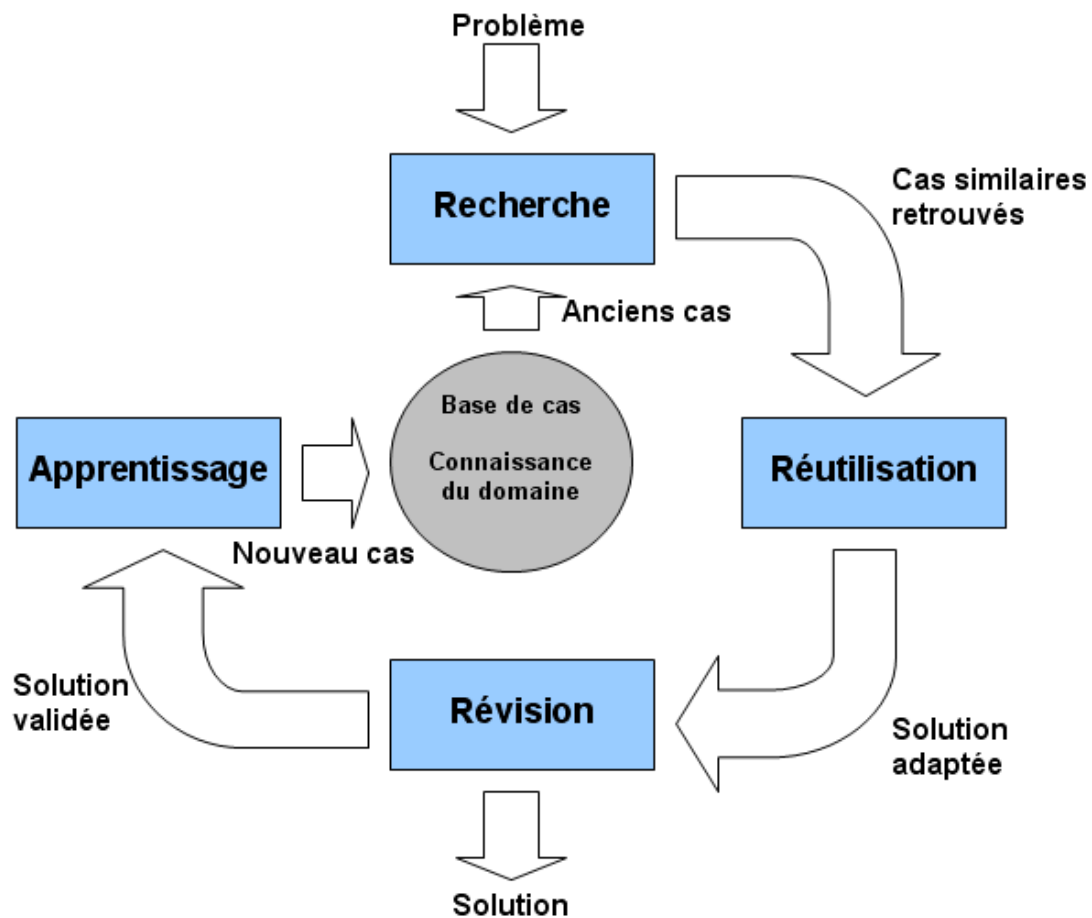


FIGURE 4.1 – Principales étapes d’un système de raisonnement à partir de cas

4.2 Évaluation des images

Le principe d’évaluation des images va permettre d’identifier clairement les types de segmentation et les paramètres à appliquer.

On peut alors distinguer deux types d’informations permettant d’évaluer une image, les informations propres à l’image et les informations propres au patient. (Informations listées par J. Henriet [6])

Informations dans l’image

On retrouve dans l’image des informations qui peuvent distinguer les séries d’images les unes des autres.

- L’intensité
- La position des pixels
- La moyenne
- La dissymétrie de l’image
- La variation des coefficients
- La centre de l’image
- La variance

- L'entropie
- Le niveau de gris et le nombre de pixels avec ce niveau de gris
- Le nombre total de pixels.

Informations du patient

On retrouve par ailleurs des informations au niveau du patient qui peuvent distinguer les séries d'images les unes des autres.

- L'âge
- Le sexe
- Le nombre de coupes de la série
- Le niveau de développement de la partie abdominale

4.3 Utilisation dans le cadre de la segmentation

Étant donné la présence d'agents dans la totalité de l'architecture, les recherches se sont dirigées vers l'apprentissage multi-agent par coopération [8], structure dans laquelle plusieurs agents collaborent pour résoudre des problèmes.

Dans ce type de structure, chaque agent utilise les raisonnements à partir de cas.

4.3.1 Premières recherches

Les premières recherches ont donc été dirigées vers l'apprentissage multi-agent par coopération. Ce système prenant place dans un système multi-agents permet à ces mêmes agents, utilisant les raisonnements à partir de cas, de communiquer entre eux afin de collaborer et ainsi trouver une solution au problème.

Une solution permet notamment la résolution du problème dans un système où les agents sont vus comme un comité. Le problème reçu par un agent est diffusé à l'ensemble des agents du système. Chacun d'eux peut alors rechercher une solution adaptée à ce problème. Ceci étant fait, les agents votent tous pour la classe de solution se rapprochant, d'après eux, le plus de la meilleure solution au problème. C'est alors la classe de solution ayant obtenu le plus de vote qui devient la solution utilisée pour résoudre le problème.

Dans l'optique de réduire le nombre de messages échangés dans le système ainsi que le temps écoulé pour trouver une réponse au problème, d'autres solutions proposent de tenter une résolution du problème par l'agent le recevant avant de le diffuser aux autres agents du système.

L'éventualité de l'utilisation de l'apprentissage par coopération a ensuite été abandonnée au profit d'un agent unique utilisant le raisonnement à partir de cas.

4.3.2 Utilisation sur un agent unique

Le choix d'un agent unique a donc été fait pour réaliser le choix automatique de segmentation à appliquer aux images. Ce choix est motivé par le fait que l'application n'a à gérer qu'un seul et même problème : *Quel type de segmentation appliquer ?*

Dans ce sens, utiliser un ensemble d'agents raisonnant à partir de cas et avec des bases de cas toutes différentes n'a pas de réel intérêt. De plus, étant donné le problème unique, toutes les bases de cas finiraient par avoir les mêmes données. Il est donc préférable de centraliser toutes les informations en une seule base de cas et ainsi accélérer la résolution du problème.

Un agent nommé *Agent CBR* sera donc utilisé et aura pour but de trouver le bon type de segmentation à appliquer aux séries de scanner.

Néanmoins, un problème naît de cette configuration de l'agent unique : comment cet agent peut-il résoudre un problème si sa base de cas ne contient aucune donnée se rapprochant de la solution optimale ?

4.3.3 Problématique de l'agent unique

La problématique de la solution introuvable vient ici du fait qu'un agent raisonnant à base de cas est capable de résoudre des problèmes grâce à sa base de cas. Cette même base de cas qu'il développera grâce à son expérience, au fil des résolutions de problèmes.

Afin de palier partiellement à ce problème, la base de cas devra donc être pré-remplie, d'une part manuellement avec des cas classiques, ou rares et atypiques, d'autre part, en lançant ce même agent sur plusieurs cas différents afin qu'il se forge sa propre expérience et renforce sa base de cas.

Enfin, le risque de tomber sur des cas d'images dont la solution ne pourrait être trouvée existe toujours. Si l'agent CBR se retrouve dans le cas où sa meilleure solution est sous le seuil critique d'optimalité, il aura alors pour objectif de lancer plusieurs segmentations aux types et paramètres différents, afin d'obtenir une batterie de solutions segmentées à évaluer.

Une fois l'ensemble des segmentations appliquées, il évaluera donc les résultats afin de n'en garder qu'un, la solution finale.

4.4 Amélioration de l'architecture

La notion de raisonnement à partir de cas étant définie dans le cadre du projet, de nouveaux agents peuvent faire évoluer l'architecture de ce dernier.

4.4.1 Partie CBR

Comme définit précédemment, l'agent CBR a pour rôle de trouver la segmentation à appliquer aux images données au logiciel afin d'obtenir le meilleur résultat possible.

Lancé directement par l'agent logiciel au lancement de l'application par l'utilisateur, cet agent va donc déterminer le ou les types de segmentations à appliquer et lancera ce processus de segmentation via l'agent *Segmentation X*.

Une fois les résultats segmentés récupérés, si plusieurs segmentation ont été appliquées, l'agent CBR lancera alors le processus d'arbitrage final afin de déterminer quel résultat retenir.

Il veillera ensuite à ce que l'arbitrage médical soit appliqué au résultat retenu et retournera l'ensemble des images modifiées à l'agent *Logiciel*.

4.4.2 Partie Segmentation X

L'agent Segmentation X est l'intermédiaire entre l'agent de *Segmentation par coupe* et l'agent *CBR*. Lancé par ce dernier, il applique la segmentation qui lui est indiquée sur les images données.

Il lance le processus de segmentation sur l'ensemble des coupes, récupère le résultat et le retourne à l'agent *CBR*.

Pour le processus de segmentation, il est lancé n fois, n étant le nombre de segmentations déterminé par l'agent *CBR* à appliquer.

Pour le processus d'arbitrage médical, l'agent est utilisé encore une fois comme intermédiaire entre l'agent de *Segmentation par coupe* et l'agent *CBR*. Il permet alors d'appliquer ledit arbitrage sur l'ensemble des coupes de la série d'images segmentées finale.

4.4.3 Partie arbitrage final

La notion d'arbitrage final intervient dans le choix final de la segmentation à garder. Le logiciel ayant appliqué plusieurs segmentations au type et aux paramètres différents, il ne peut en garder qu'un.

Ainsi, le module d'arbitrage final va permettre l'évaluation des résultats de chaque segmentation afin d'en retirer la plus optimale.

Une fois ce choix effectué, l'application pourra lancer l'arbitrage médical sur l'ensemble des coupes de la série segmentée sélectionnée et ainsi récupérer toutes les images modifiées finales à mettre à disposition de l'utilisateur.

4.5 Architecture finale

Dans les images suivantes (Fig 4.2,4.3 et 4.4, 4.5), l'architecture de l'application est décrite à l'aide de diagrammes UML de séquence et d'activité.

On y retrouve les principes décrits précédemment.

1. L'utilisateur fournit les séries de DICOM en temps tardif et vasculaire au logiciel
2. L'utilisateur lance l'application
3. Le logiciel lance l'agent CBR
4. L'agent CBR détermine les segmentations optimales à appliquer
5. L'agent CBR lance le processus de segmentation
6. L'agent CBR lance l'agent de segmentation X en lui fournissant le type et les paramètres de la segmentation n
7. L'agent segmentation X lance le processus de segmentation des images sur la première coupe
8. La segmentation X lance l'agent de segmentation par coupe en lui fournissant la coupe z de DICOM
9. L'agent de segmentation par coupe lance la segmentation sur les DICOM de la coupe z
10. La segmentation des deux DICOM de la même coupe est parallélisée
11. L'agent de segmentation par coupe récupère les DICOM segmentés
12. Si l'ensemble des coupes n'a pas été traité, on retourne à l'étape 8 avec le z suivant
13. Sinon l'agent segmentation X a terminé, récupère les DICOM segmentés et les retourne à l'agent CBR
14. Si l'ensemble des segmentations n'a pas été traité, on retourne à l'étape 6 avec le n suivant
15. Sinon l'agent CBR a terminé l'exécution des segmentations
16. L'agent CBR lance l'agent d'arbitrage final
17. La segmentation optimale est trouvée et une des séries de DICOM segmentés est gardée, l'agent CBR est averti
18. L'agent CBR lance l'arbitrage médical sur la série des DICOM segmentés sélectionnée
19. L'arbitrage médical est appliqué sur l'ensemble des coupes de la série et le résultat est retourné
20. L'agent CBR récupère l'image arbitrée et la retourne au logiciel avec les DICOM segmentés

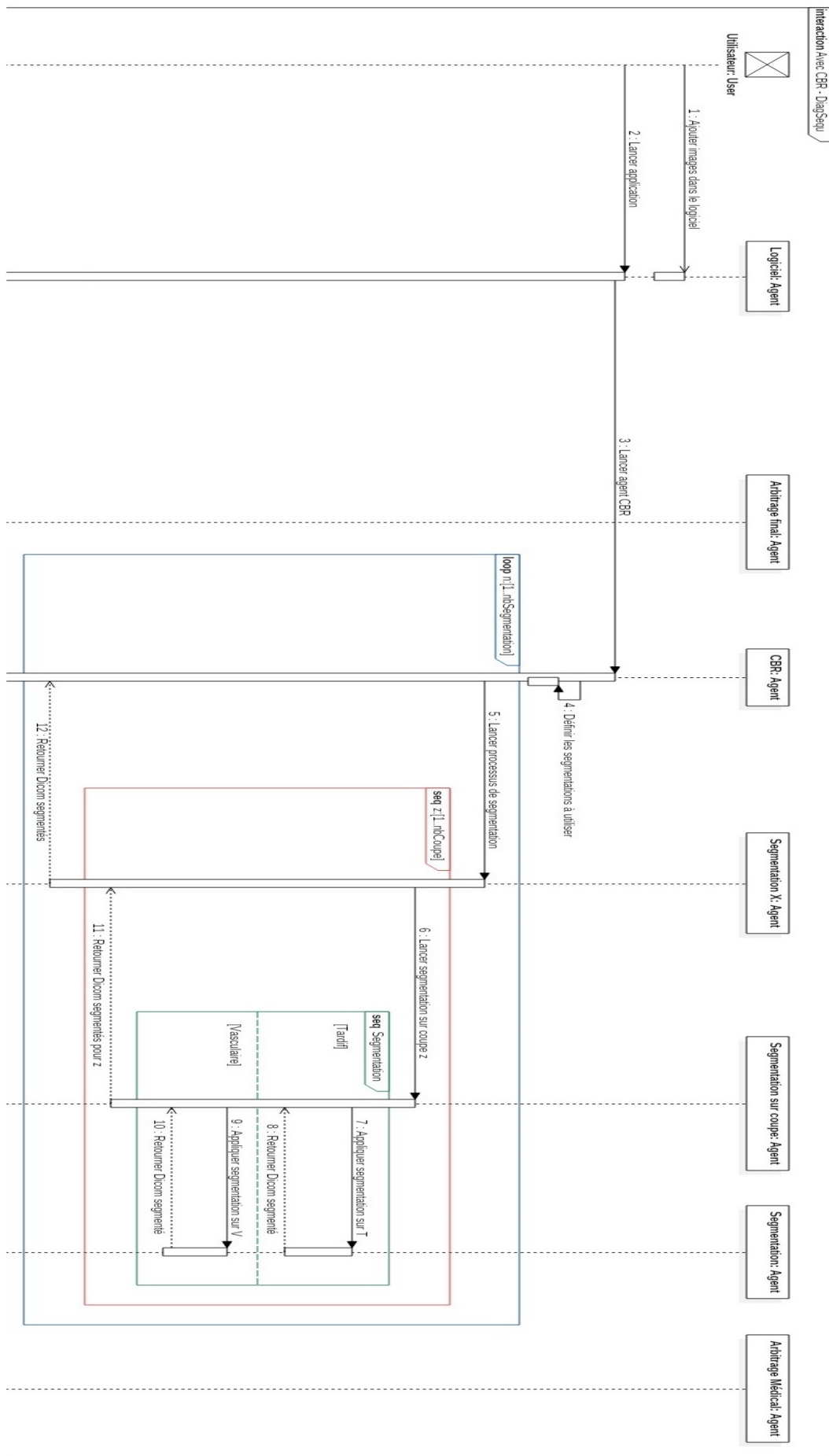


FIGURE 4.2 – Diagramme de séquence - Architecture du système - Partie 1

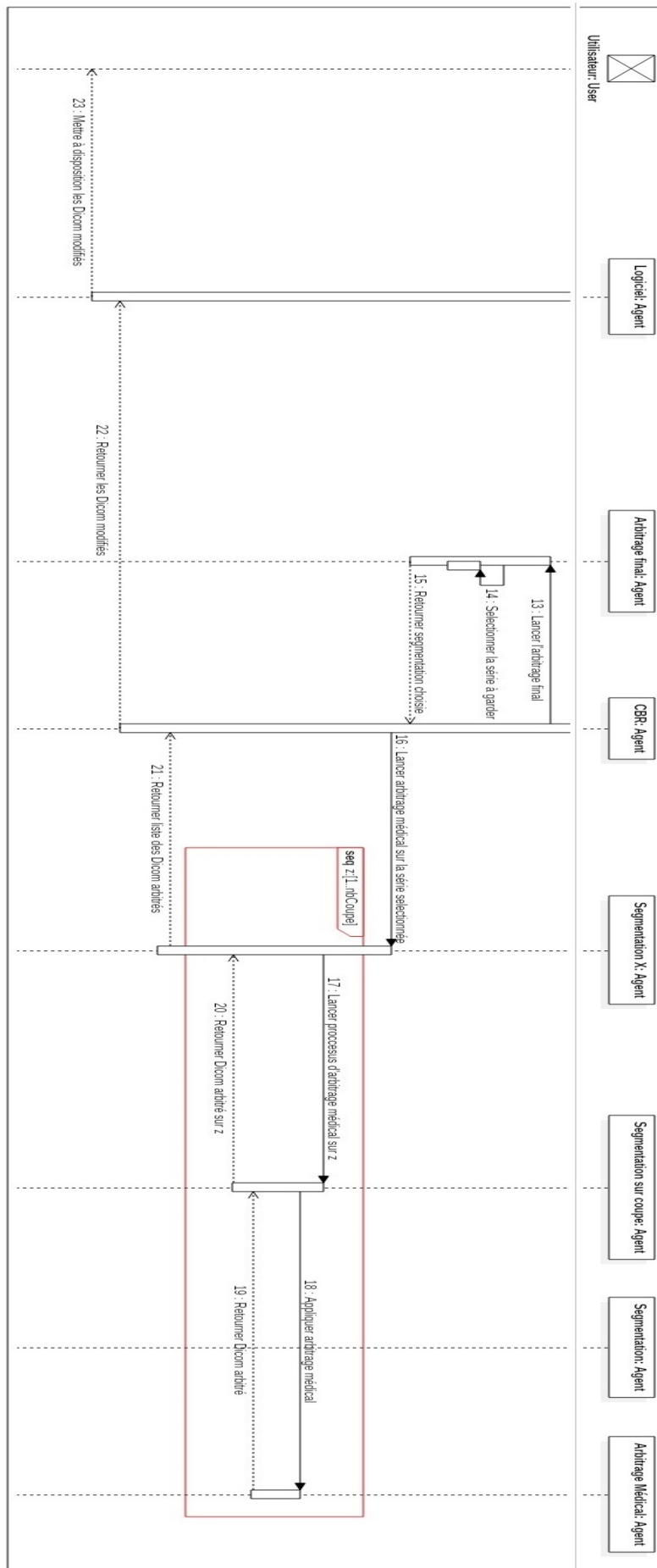


FIGURE 4.3 – Diagramme de séquence - Architecture du système - Partie 2

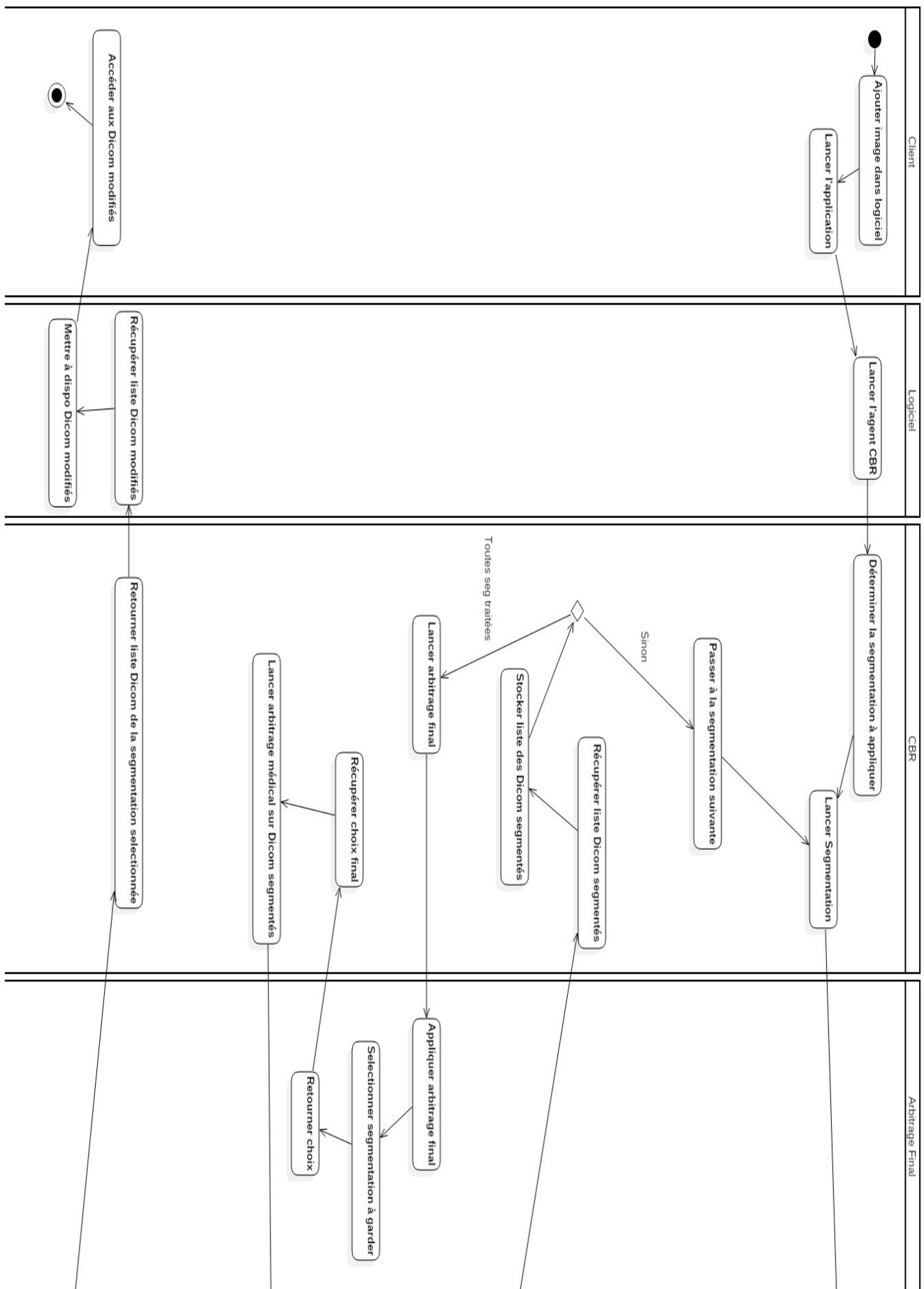


FIGURE 4.4 – Diagramme d'activité - Architecture du système - Partie 1

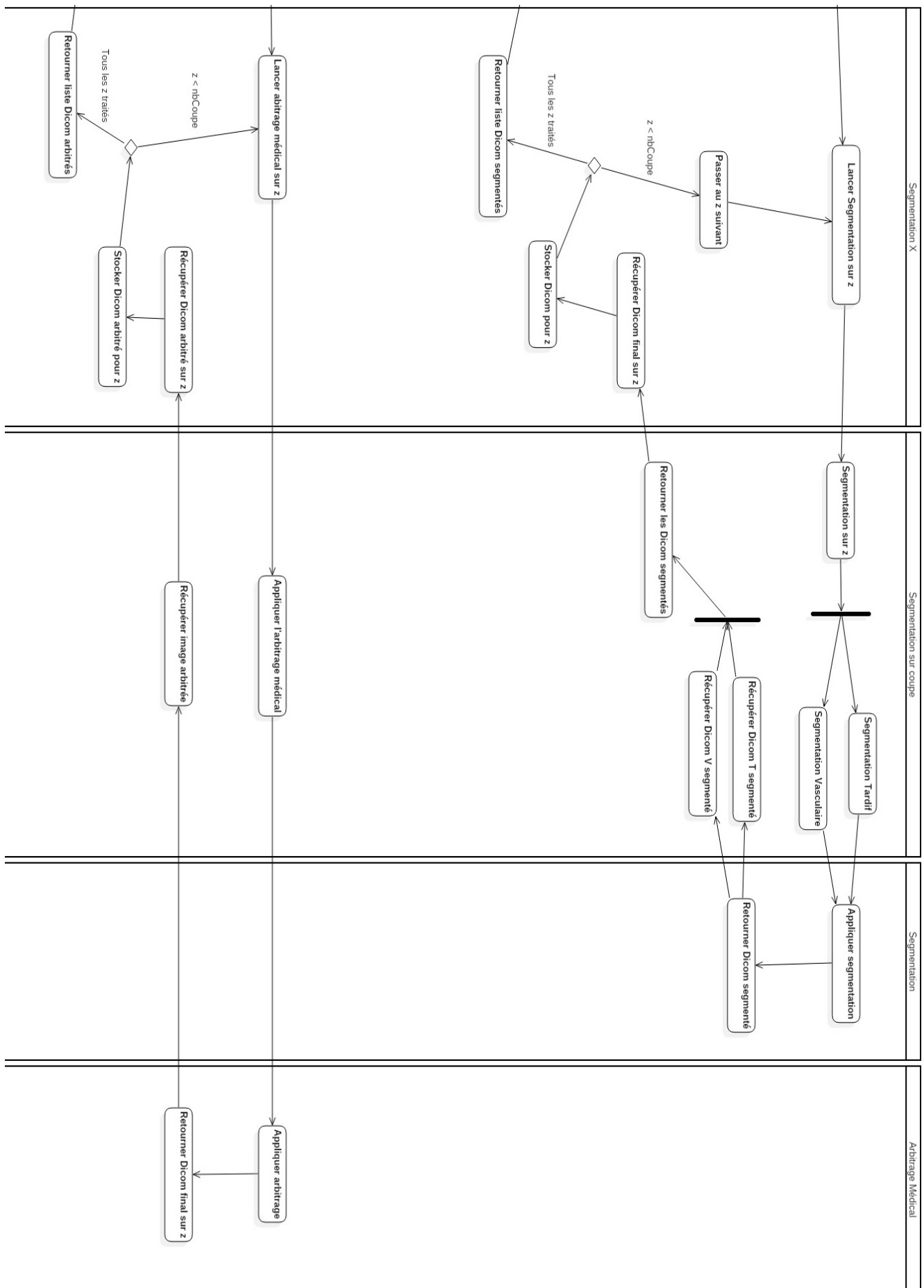


FIGURE 4.5 – Diagramme d'activité - Architecture du système - Partie 2

Chapitre 5

Conclusion

5.1 Bilan personnel

Ce projet a été pour nous l’occasion de mettre en œuvre la réalisation d’un projet de recherche novateur.

Par ailleurs nous aurons eu l’occasion de découvrir l’univers du médical et ses problématiques informatiques.

Concernant l’avancement du projet, l’architecture de l’application a été totalement modélisée.

Notre volonté de nous investir davantage était réelle mais le manque cruel de ressources sur lesquelles travailler a largement contribué à l’absence du développement de modules comme l’arbitrage médical.

Les technologies déjà définies et les ébauches de travail concernant la programmation de l’arbitrage médical pourront néanmoins servir de base de travail et orienter les futurs développeurs en charge du projet.

5.2 Perspectives

Le but de ce projet était de déterminer des bases solides afin d’entamer le développement du projet SAIAD.

Les prochaines personnes en charge du projet pourront utiliser notre architecture et nos recherches comme base de développement et ainsi gagner du temps.

L’arbitrage médical, fonctionnalité encore inexistante à ce jour, peut être développé à partir des recherches effectuées et des exigences définies.

Table des figures

1.1	Exemple d'image DICOM anonymée	5
1.2	Exemple d'image scanner sans injection	7
1.3	Exemple d'image scanner avec injection au temps vasculaire	8
1.4	Exemple d'image scanner avec injection au temps tardif	8
2.1	Diagramme de séquence - Architecture "ultra" globale du système	9
2.2	Diagramme de séquence - Architecture du système	13
2.3	Diagramme d'activité - Architecture du système	14
3.1	Exemple d'arbitrage médical	17
3.2	Emplacement des reins par rapport à la vertèbre L2	19
3.3	Diagramme d'activité - Arbitrage médical	20
4.1	Principales étapes d'un système de raisonnement à partir de cas	22
4.2	Diagramme de séquence - Architecture du système - Partie 1	26
4.3	Diagramme de séquence - Architecture du système - Partie 2	27
4.4	Diagramme d'activité - Architecture du système - Partie 1	28
4.5	Diagramme d'activité - Architecture du système - Partie 2	29
5.1	Diagramme de classe - Arbitrage médical	33
5.2	Diagramme de classe - Architecture sans CBR	34
5.3	Diagramme de classe - Architecture avec CBR	35

Bibliographie

- [1] Format DICOM.
https://fr.wikipedia.org/wiki/Digital_imaging_and_communications_in_medicine
- [2] Système Multi-Agents.
https://fr.wikipedia.org/wiki/Systeme_multi-agents
- [3] Raisonnement à partir de cas.
https://fr.wikipedia.org/wiki/Raisonnement_par_cas
- [4] Imebra - Librairie de manipulation des fichiers DICOM.
<https://imebra.com/>
- [5] Opencv - Librairie de traitement d'image.
<http://opencv.org/>
- [6] J. Henriët. Etat de l'art sur la segmentation d'images par Raisonnement à Partir de Cas.2015
- [7] Janet KOLODNER. Case-based reasoning. CA Morgan Kaufmann. 1993.
- [8] Enric Plaza and Santiago Ontañón. Cooperative Multiagent Learning.

Annexes

5.3 Diagrammes de classe

5.3.1 Arbitrage médical

Agent Segmentation sur coupe
-SegmentationType -Paramètres -DicomT -DicomV
+LancerSegmentation(dicom): DicomSegmenté +LancerArbitrage(Tsegmenté, Vsegmenté): DicomArbitré

Agent Arbitrage Médical
-DicomT segmenté -DicomV segmenté -DicomArbitré -tabROI
+LancerArbitrage(DicomT, DicomV): DicomArbitré -AppliquerArbitrage() -GetROI(idROI): tabCoordROI -TransfertROI(idROI) -Recalibrer() -DétecterROI()

FIGURE 5.1 – Diagramme de classe - Arbitrage médical

5.3.2 Architecture sans CBR



FIGURE 5.2 – Diagramme de classe - Architecture sans CBR

5.3.3 Architecture avec CBR

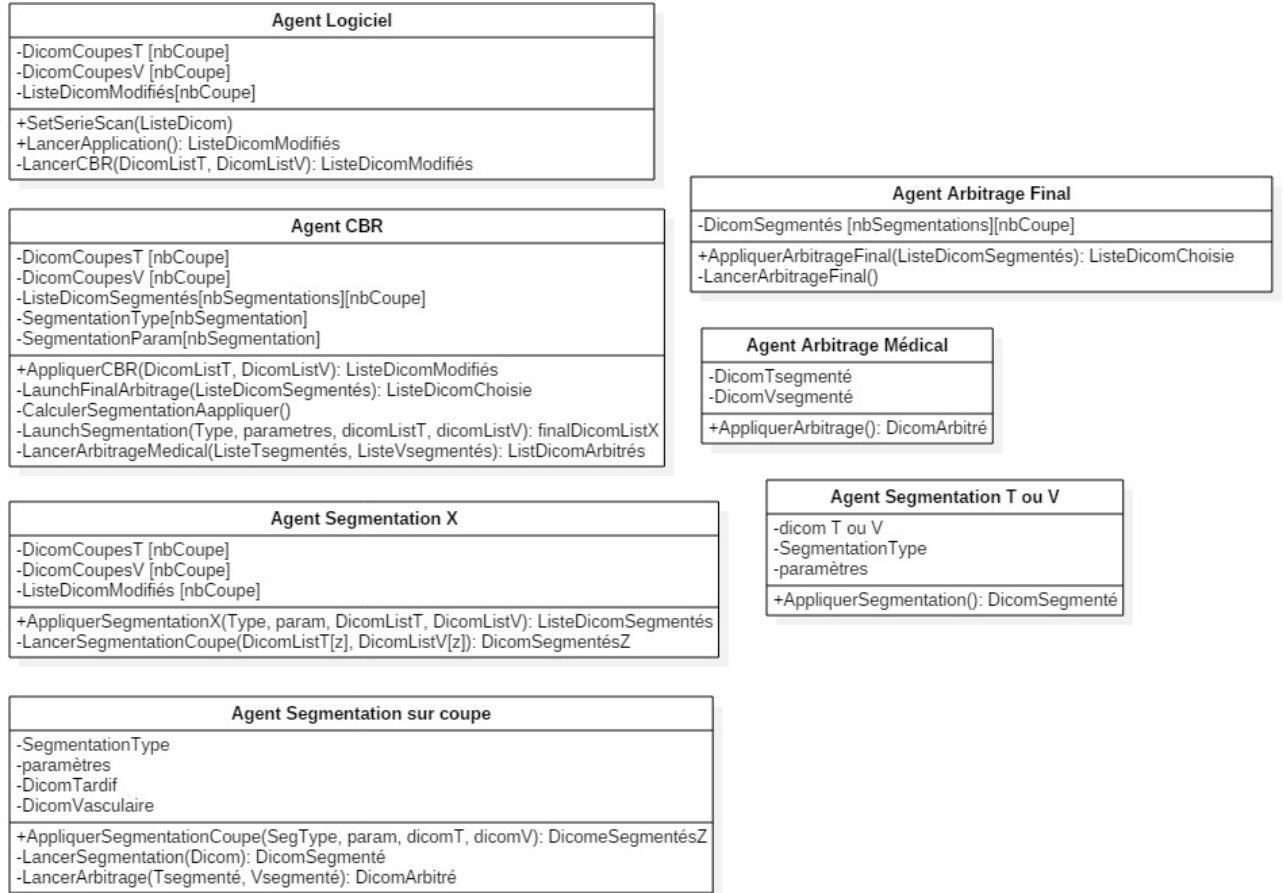


FIGURE 5.3 – Diagramme de classe - Architecture avec CBR

Résumé

Le projet SAIAD a pour objectif d'aider les médecins dans le traitement des enfants atteints du cancer du rein.

Le diagnostic de ce type de cancer est basé sur le traitement d'image et la représentation 3D de la tumeur.

Ce processus requérant l'intervention humaine est extrêmement chronophage.

L'objectif est donc de réaliser un système capable de réaliser ces tâches automatiquement.

Ce projet a pour but de réaliser l'architecture distribuée du système de base de l'application.

L'application, totalement autonome est composée de modules de segmentation, d'arbitrage médical et de modélisation 3D.

Le système est capable d'appliquer les traitements attendus et de mettre à disposition les résultats obtenus.

Mots clés : Cancer du rein, projet SAIAD, Segmentation automatique, Raisonnement partir de cas, Arbitrage médical.

Abstract

The objective of the SAIAD project is to help medical doctors in the treatment of children affected by the renal cancer.

The diagnostic of such cancer is based on image processing and tri-dimensional representations of the tumor.

However, this process still requires human interventions and is time-consuming.

Our objective is to provide a system that will be able to provide such numerical representations automatically.

This project aims at implementing a distributed architecture based on autonomous and intelligent processes (agents) that will combine different processes in order to provide tri-dimensional representation of tumors taking the patient scans as input parameters.

The agents will implement case-based reasoning in order to optimise the process combinations and parameters.

We designed and modeled this multi-agent system.

Keywords : Renal cancer, SAIAD project, Automatic segmentation, case-based reasoning, Medical policy.