

Master 1 Informatique Professionnel et Recherche Unifié

Réalisation d'un logiciel de Visioconférence

Cahier des charges

Version 2.0 du 06/02/08 envoyée à M. Michel Leclère,
responsable du Master 1 Informatique



Réalisé par :

ESCANDELL Frédéric, NOUAILLES Yoann, MORA Michaël et QUINTON Clément

Projet encadré par Anne-Elisabeth BAERT et Vincent BOUDET

Année universitaire 2007-2008

Table des matières

1	Contexte	3
2	Organisation du projet	4
2.1	Planning général du projet	4
2.1.1	Élaboration du cahier des charges	5
2.1.2	Étude de l'existant	5
2.1.3	Analyse et conception	5
2.1.4	Développement	6
2.1.5	Déploiement de l'application	6
2.1.6	Tests	6
2.2	Organisation du groupe	7
2.2.1	Logiciel de gestion de projet et versionnage	7
2.2.2	Rythme des réunions au sein du groupe et avec nos tuteurs	7
2.2.3	Division des tâches	8
3	Spécifications générales	9
3.1	Présentation générale	9
3.2	Exigences sur la capacité du logiciel	9
3.2.1	Fonctionnalités nécessaires pour la visioconférence	10
3.2.2	Fonctionnalités nécessaires pour le «tableau blanc»	10
3.2.3	Contraintes sur la sûreté du logiciel	10
3.2.4	Contraintes sur l'environnement du logiciel	11
3.2.5	Contraintes de conception	11
3.2.6	Facteurs de qualité du logiciel	11
4	Bibliographie	12

Table des figures

2.1	Diagramme de Gantt.	4
3.1	Croquis de l'interface.	9
3.2	Exemple de VPN d'accès.	11

1 Contexte

Le but de ce TER est de développer un logiciel de visioconférence interactif. Cet outil doit permettre d'établir une communication entre un radiologue et un patient lorsque celui-ci se trouve chez son médecin. Ce dialogue doit permettre au radiologue d'éclairer les patients sur leurs interrogations et de fournir une explication plus précise au médecin l'accompagnant, grâce au support radio numérique.

Cet outil se doit donc d'être ergonomique et facile d'utilisation. De nombreux logiciels sont disponibles et pourraient éventuellement répondre en partie aux contraintes posées par le sujet, c'est pourquoi ce TER débutera par une étude approfondie/détaillée de l'existant.

2 Organisation du projet

2.1 Planning général du projet

L'illustration ci-dessous présente les différentes phases du cycle de vie en V (i.e. toute étape doit être achevée avant de passer à l'étape suivante) de notre application de visioconférence.

ID	Nom de tâche	Début	Terminer	Durée	jan. 2008		févr. 2008				mars 2008					avr. 2008			mai 2008				
					20/1	27/1	3/2	10/2	17/2	24/2	2/3	9/3	16/3	23/3	30/3	6/4	13/4	20/4	27/4	4/5			
1	Rédaction cahier des charges	16/01/2008	01/02/2008	13j	[Barre bleue]																		
2	Etude de l'existant	28/01/2008	11/02/2008	11j	[Barre bleue]																		
3	Analyse Conception	11/02/2008	13/02/2008	6j	[Barre bleue]																		
4	Developpement	18/02/2008	07/04/2008	36j	[Barre bleue]																		
5	Dépolement de l'application	07/04/2008	14/04/2008	6j	[Barre bleue]																		
6	Tests de l'application	10/04/2008	14/05/2008	25j	[Barre bleue]																		
7	Rédaction rapport, préparation oral	07/04/2008	23/04/2008	13j	[Barre bleue]																		

FIG. 2.1 – Diagramme de Gantt

2.1.1 Élaboration du cahier des charges

Pour concevoir au mieux notre cahier des charges, plusieurs réunions tuteurs/étudiants ont été nécessaires, afin de définir clairement leurs attentes (qui sont en fait celles des professionnels utilisant le logiciel) concernant le projet. La première d'entre elles avait comme objectif de clarifier les grandes lignes du projet, telles qu'elles avaient été définies par l'entreprise. Les suivantes nous ont permis d'éclaircir certains points restés flous lors de la définition du sujet.

Diverses réunions entre tous les éléments du groupe ont également été programmées, afin de confronter nos idées et d'en déduire l'organisation des tâches. La synthèse de ces réunions a conduit à la construction de ce cahier des charges lequel, suite à une première version mise en page sous Open Office, a ensuite été généré en L^AT_EX.

2.1.2 Étude de l'existant

Le concept de logiciel de visioconférence est déjà largement représenté sur le marché. Nous allons donc nous intéresser à plusieurs d'entre eux pour réaliser un inventaire des fonctionnalités intéressantes à mettre en oeuvre. Cette étude préliminaire nous offrira la possibilité soit de développer notre propre application, soit de partir sur une base existante. Ce choix réalisé, nous pourrons ensuite intégrer des modules déjà existants dans notre logiciel et/ou en supprimer d'autres selon nos besoins.

2.1.3 Analyse et conception

La modélisation des différentes parties de notre application se fera en UML (Unified Modeling Language) et tiendra compte des différentes fonctionnalités et contraintes associées au logiciel. UML met à notre disposition plusieurs diagrammes permettant d'analyser les besoins du client (diagramme de cas d'utilisation, diagramme de séquence système) jusqu'à la modélisation détaillée (diagramme de classes de conception). La conception sous-jacente maximisera la réutilisabilité et l'évolutivité de notre logiciel grâce notamment à l'utilisation de patrons de conception (design patterns) si nécessaire.

2.1.4 Développement

Le développement de notre application devra respecter un certain nombre de conventions que nous mettrons en place au sein du groupe. Le respect de ces conventions assurera l'homogénéité du code et permettra à toute l'équipe de comprendre et de maintenir facilement le code écrit par une tierce personne. Par exemple, si le langage retenu pour le développement est Java nous utiliserons JavaStyle ¹. D'autre part nous mettrons en place sur un serveur un logiciel de gestion de projet et versionnage SVN (SubVersioN) ainsi qu'un logiciel de suivi de projet TRAC.

2.1.5 Déploiement de l'application

Cette partie consistera à créer une version du logiciel installable et exécutable sur toutes les plateformes demandées par le client. Nous utiliserons plus particulièrement Windows qui est le système d'exploitation le plus communément utilisé par les médecins, sans toutefois exclure les systèmes Linux.

2.1.6 Tests

Une phase de tests dans les conditions de son utilisation future devra être réalisée. Cette phase sera appuyée d'une étude de performances. Une batterie de tests automatiques couplée à des tests manuels sera également effectuée. Enfin, les tests porteront sur la portabilité de notre application sur les différents systèmes d'exploitation.

¹Conventions de nommage édictées par Sun.

2.2 Organisation du groupe

2.2.1 Logiciel de gestion de projet et versionnage

Trac est un système de gestion de projet logiciel et de suivi de défauts / bogues via le web. Il propose une interface pour le système de contrôle de source Subversion, un Wiki intégré, et un nombre intéressant d'option permettant de rester au courant des événements et changements du projet. Trac inclus un Wiki : c'est un système de gestion de contenu de pages Web modifiables par tous les visiteurs autorisés. Cela facilite l'écriture collaborative de documents avec un minimum de contraintes.

Toutes les pages Wiki sont éditables via le navigateur web si vous êtes authentifié et possédez les droits nécessaires. Un historique permet de visualiser et d'accéder rapidement aux dernières modifications du projet, classées par date. Le logiciel Trac propose également un explorateur Subversion, qui permet de naviguer dans le code source du logiciel à partir d'un navigateur web.

2.2.2 Rythme des réunions au sein du groupe et avec nos tuteurs

Tous les étudiants de ce groupe étant dans le même cursus, l'emploi du temps nous laisse assez de disponibilités pour faire des réunions régulières et travailler ensemble. D'autre part nous avons défini une réunion bimensuelle avec nos tuteurs afin de faire un point sur l'avancement du projet et valider nos choix. Ce moment pourra être utilisé pour interagir avec la société E-R Expertise-Radiologie, initiatrice du projet.

2.2.3 Division des tâches

A l'heure actuelle, la division des tâches s'est limitée à l'étude de l'existant.

Etudiant	Logiciels étudiés
ESCANDELL Frédéric	Clients Java réseau Jabber : JBrother API JAVA pour client Jabber : Smack API 3.0.4
NOUAILLES Yoann	VLC
MORA Michaël	Openwengo
QUINTON Clément	Windows Live Messenger Skype

Nous avons choisi de fonctionner de la manière suivante : chaque membre du groupe se focalise sur l'étude d'un ou plusieurs outils existant(s) et le(s) présente aux autres membres du groupe.

Cette manière de procéder nous permet d'avoir une vision plus globale de l'état d'art et d'envisager une possible intégration d'un outil existant. Une fois cette étude terminée et les choix retenus pour la conception, nous pourrions établir une nouvelle division des tâches.

3 Spécifications générales

3.1 Présentation générale

Ce logiciel permet de réaliser une visioconférence entre deux personnes. Ces personnes peuvent se voir mutuellement et converser via un support audio. Une fonctionnalité supplémentaire permettra aux docteurs de réaliser des annotations en temps réel sur une radiographie numérique visible par les deux parties.

3.2 Exigences sur la capacité du logiciel

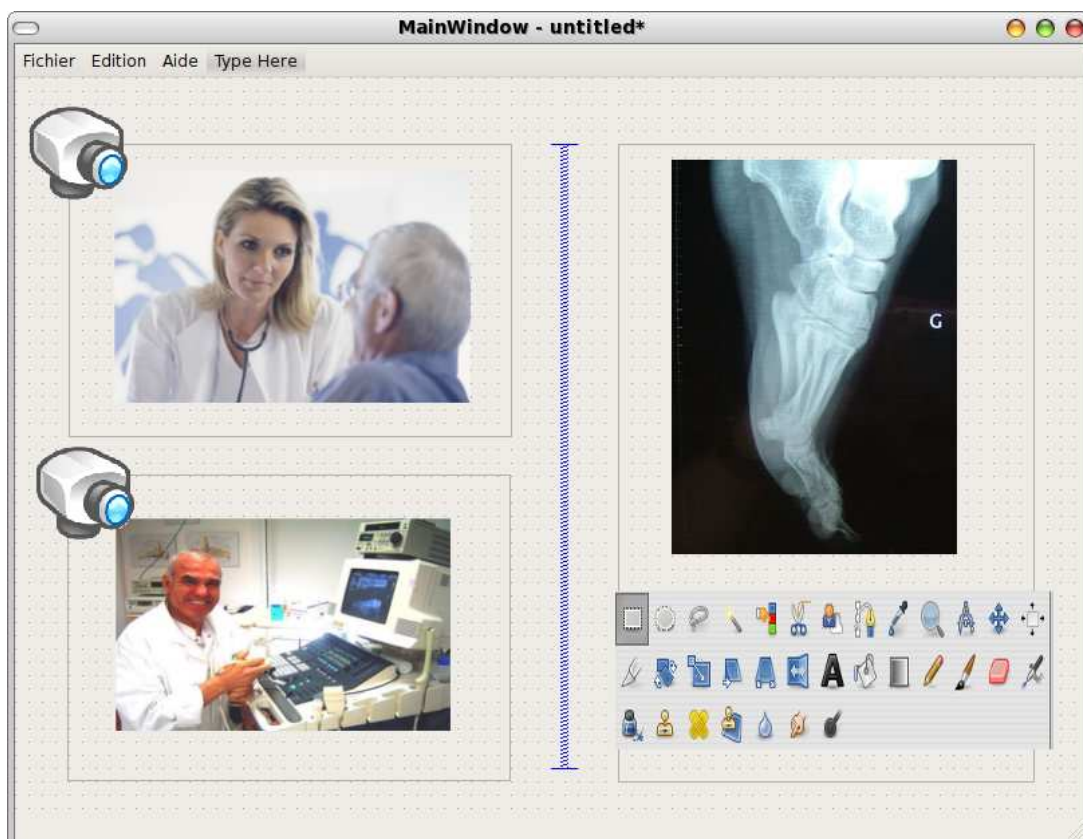


FIG. 3.1 – Croquis de l'interface

3.2.1 Fonctionnalités nécessaires pour la visioconférence

Le logiciel permet à deux personnes distantes de converser comme représenté dans l'illustration ci-dessus. Notre interface comprendra une partie dédiée à l'affichage des webcams correspondant au médecin et au radiologue. Elle doit également permettre une communication audio entre les deux interlocuteurs. Par exemple, un médecin à Pézénas peut demander le diagnostic d'un radiologue situé au CHU de Montpellier concernant son patient.

L'utilisateur dispose d'une liste de contacts et peut décider de lancer une visioconférence. L'invitation doit être acceptée par l'utilisateur destinataire pour démarrer la visioconférence. Chaque utilisateur doit avoir la possibilité de lancer plusieurs conversations privées (afin de conserver la confidentialité des patients) en parallèle.

Concernant l'interface du logiciel et d'après l'étude réalisée, nous avons pensé qu'il serait préférable d'implémenter deux types d'interface différents : une simpliste pour le médecin traitant lui permettant de communiquer avec le radiologue, une plus élaborée autorisant le radiologue à communiquer avec plusieurs médecin en laissant en attente ceux ne faisant pas partie de la communication courante. Ainsi, chaque IHM (Interface Homme Machine) serait adaptée au besoin de l'utilisateur.

3.2.2 Fonctionnalités nécessaires pour le «tableau blanc»

Notre application fournit un panel d'outils pour réaliser des annotations sur une radiographie préalablement chargée par les deux utilisateurs. Les modifications effectuées sur la radiographie sont envoyées en temps réel à l'utilisateur avec lequel on converse.

3.2.3 Contraintes sur la sûreté du logiciel

Le logiciel doit fonctionner sur le réseau VPN (Virtual Private Network) afin de préserver les données confidentielles échangées entre les médecins et leurs patients. Un réseau VPN repose sur un protocole appelé "protocole de tunneling" (Figure 3.2).

Ce protocole permet de faire circuler les informations de l'entreprise de façon cryptée d'un bout à l'autre du tunnel. Ainsi, les utilisateurs ont l'impression de se connecter directement sur le réseau de leur entreprise.

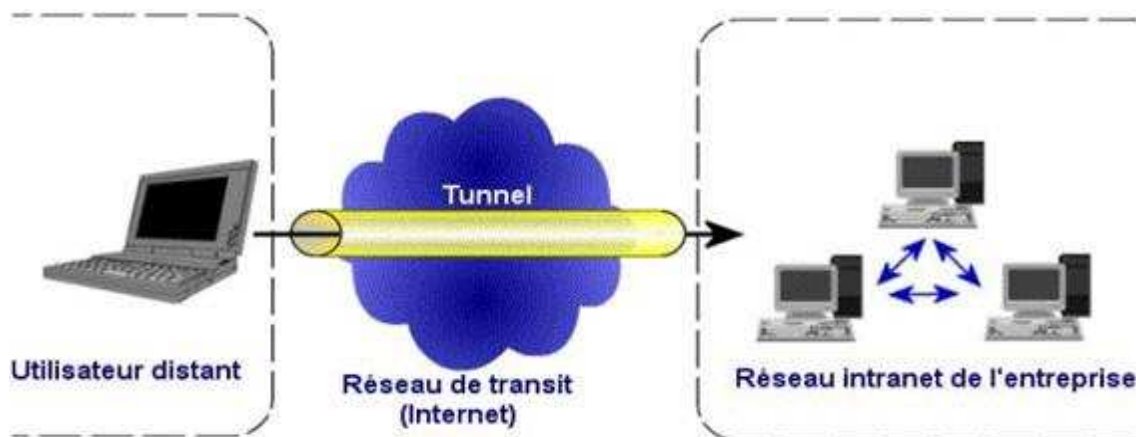


FIG. 3.2 – Exemple de VPN d'accès

Le principe de tunneling consiste à construire un chemin virtuel après avoir identifié l'émetteur et le destinataire. Par la suite, la source chiffre les données et les achemine en empruntant ce chemin virtuel.

3.2.4 Contraintes sur l'environnement du logiciel

Le logiciel sera développé afin de fonctionner sous Windows majoritairement et sous Linux éventuellement. Nous privilégierons en effet les systèmes Windows Xp et Windows Vista, qui sont ceux utilisés par les médecins qui se serviront du logiciel. Cependant, il serait préférable d'obtenir un logiciel portable sur tout type de plateformes, vue la place grandissante prise par le logiciel libre dans le monde de l'informatique.

3.2.5 Contraintes de conception

Le logiciel devra s'intégrer facilement dans une future application web que développera la société. Il serait intéressant de prendre en compte dans la mesure du possible le type de machine sur laquelle tournera l'application (mono ou bi-cœur, 32 ou 64 bits), afin d'en tirer le meilleur profit possible en terme de performances, optimisation, etc...

3.2.6 Facteurs de qualité du logiciel

La société nous demande un logiciel convivial, c'est à dire un logiciel offrant une prise en main simple et rapide et au graphisme intuitif et agréable (limiter le nombre de boutons et d'options, ne pas surcharger l'interface graphique). Ce logiciel sera en effet utilisé par de nombreuses personnes y compris des personnes âgées connaissant très peu l'informatique.

4 Bibliographie

Références diverses

- JavaStyle : <http://java.sun.com/docs/codeconv/CodeConventions.pdf> ;
- Trac : <http://trac.edgewall.org/> ;
- E-R Expertise-Radiologie : <http://expertise-radiologie.com/default.aspx> ;
- VPN : <http://www.frameip.com/vpn/> ;

Logiciels étudiés

- JBrother : <http://www.jbrother.org/index.rb?command=screenshots> ;
- Smack API 3.0.4 : <http://www.igniterealtime.org/projects/smack/index.jsp> ;
- VLVLC : <http://www.vlvc.net/fr-downloads.html> ;
- Openwengo : <http://openwengo.orgopenwengo.org> ;
- Windows Live Messenger : <http://www.windowlive.fr/messenger/default.asp> ;
- Skype : <http://www.skype.com/intl/fr/> ;