Refonte d'un système de vote interactif

Romain Bernardi

▶ **To cite this version:**

Romain Bernardi. Refonte d'un système de vote interactif. Génie logiciel [cs.SE]. 2012. dumas-01063226

HAL Id: dumas-01063226

https://dumas.ccsd.cnrs.fr/dumas-01063226

Submitted on 18 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS
CENTRE REGIONAL D'AUVERGNE

MEMOIRE

présenté en vue d'obtenir

le DIPLOME d'INGENIEUR CNAM

SPECIALITE : Informatique
OPTION : ISI (Informatique, Systèmes d'information)

par

BERNARDI Romain

Sujet : Refonte d'un système de vote interactif

Soutenu le 13 Juin 2012

JURY

Président : M. Kamel Barkaoui
Membres : M. Olivier Dusserre
M. Michel Misson
M. Jean-Marc Nermond
M. Michel Schneider
M. Farouk Toumani

Table des Matières

Remerciements 6

Liste des abréviations 7

Concepts métier 8

1 Introduction 9

2 Next Media 10

2.1 Son histoire 10

2.2 Situation géographique 11

2.3 Le personnel 11

2.4 L'organigramme 12

2.5 Présentation générale des activités de l'entreprise 13

2.5.1 Formation informatique 13

2.5.2 Développement d'applications 13

3 La solution QuizzBox 14

3.1 Qu'est-ce que QuizzBox 14

3.2 Fonctionnement de QuizzBox 15

3.3 QuizzBox standard 19

3.4 Les différents outils 19

3.4.1 QuizzBox 20

3.4.2 QuizzBox Solution 22

3.5 Les différentes versions 25

3.6 L'équipe QuizzBox et son fonctionnement 27

4 Nos secteurs d'activité, la concurrence 29

4.1 Le secteur de la sécurité 29

4.2 Le secteur de l'éducation 30

4.3 Les autres secteurs d'activité 30

4.3.1 Les formations interactives 30
4.3.2 Les tests automatisés 30
4.3.3 L'évaluation et la mesure 30
4.3.4 Les réunions interactives 31
4.3.5 Les Serious Games, les séminaires de motivation, lancements de nouveaux produits, réunions de fin d'année 31
4.3.6 Les animations interactives de stand 31
4.3.7 Les études de marché 32
4.3.8 Les réunions d'assemblées - Conseils municipaux 32

4.4 Nos points forts 32
4.4.1 Échanges et interactions 32
4.4.2 Gain de temps 32
4.4.3 Pédagogie interactive 33
4.4.4 Sollicitation de l'élève 33

4.5 Nos faiblesses 33
4.5.1 Le matériel 33
4.5.2 Les moyens financiers 34

5 Les nécessités d'une nouvelle version 35

5.1 Les objectifs de Bureau Veritas 36
5.1.1 Améliorer la réactivité dans les échanges de documents 36
5.1.2 Ajouter de la sécurité et du contrôle dans l'utilisation de QuizzBox 37
5.1.3 Avoir un environnement spécifique CACES 37
5.1.4 Limiter l'utilisation de l'application suivant le profil 37
5.1.5 Communiquer avec les outils de gestion d'affaires Bureau Veritas 37

5.2 Nos propres objectifs 38
5.2.1 Repenser l'architecture 38
5.2.2 Gestion de l'unicité des personnes 39
5.2.3 Gestions des examens 39
5.2.4 Gestion des épreuves pratiques 39
5.2.5 Récepteur multi-applications 40

5.3 Critique de l'architecture de QB_Utilisateur version 4 40

6 Conception de la nouvelle version, QuizzBox v5 42

6.1 La sphère métier 42

6.1.1 Cas d'utilisation 42
6.1.2 Problématique 42
6.1.3 Réflexion et solution 43

6.2 Les interfaces 46
6.2.1 Cas d'utilisation 46
6.2.2 Problématique 47
6.2.3 Réflexion et solution 48

6.3 QuizzBox en version multi-langues 55
6.3.1 Cas d'utilisation 55
6.3.2 Problématique 56
6.3.3 Réflexion et solution 57

6.4 Gestion des participants 59
6.4.1 Cas d'utilisation 59
6.4.2 Problématique 60
6.4.3 Réflexion et solution 60

6.5 Les examens 62
6.5.1 Cas d'utilisation 62
6.5.2 Problématique 62
6.5.3 Réflexion et solution 63

6.6 Les épreuves pratiques 68
6.6.1 Cas d'utilisation 68
6.6.2 Problématique 68
6.6.3 Réflexion et solution 69

6.7 Le récepteur devient multi-applications 71
6.7.1 Cas d'utilisation 71
6.7.2 Problématique 72
6.7.3 Réflexion et solution 72

7 Mise en œuvre et déploiement 74

7.1 Cahier des charges 74

7.2 Analyse 74

7.3 Développement 75

7.4 Tests 75

7.5 Déploiement 77

7.6 Relations avec le client 79

8 Evolutions futures 80

8.1 Le projet Bureau Veritas 80

8.1.1 Point sur la phase 1 80

8.1.2 La phase 2 80

8.1.3 Maintenance 81

8.2 Les autres projets 81

Conclusion 83

ANNEXE 1 – Le groupe AGINIUS 85

ANNEXE 2 – Agréments et certifications de formation 87

ANNEXE 3 – Le matériel QuizzBox 88

ANNEXE 4 – Les modules de QuizzBox 89

ANNEXE 5 – Classes métier (Session/Personnes) 98

ANNEXE 6 – Classes métier (Tests) 99

ANNEXE 7 – Classes métier (Résultats) 100

ANNEXE 8 – Classes de dé-sérialisation des interfaces 101

ANNEXE 9 – Classes de dé-sérialisation des examens 102

ANNEXE 10 – Classes de dé-sérialisation des épreuves pratiques 103

ANNEXE 11 – Classes de gestion des interfaces 104

ANNEXE 12 – Schéma de la base de données 105

Bibliographie 106

Liste des figures présentées dans le mémoire 107

Remerciements

Je tiens à remercier,

Michel Schneider, professeur à l'université de Clermont-Ferrand et mon tuteur sur ce projet, pour ses conseils précieux et le suivi fourni tout au long de la rédaction du mémoire ;

Jean-Marc Nermond, gérant de Next Media, pour m'avoir fait confiance en me laissant conduire ce projet, qui représente beaucoup pour la société ;

Olivier Dusserre, chef de projet QuizzBox, mon référant technique, pour le temps passé ensemble à confronter nos avis, ses conseils et sa gestion de l'équipe sans faille ;

Tous les enseignants du CNAM qui ont permis l'aboutissement de ma formation ;

Ma femme et mes enfants, pour leur amour indéfectible et pour m'avoir permis de passer tout ce temps ces dernières années et surtout ces derniers mois loin d'eux, sans qu'il ne me soit fait de reproches.

Liste des abréviations

Abréviation	Signification
ASSR	Attestations Scolaires de Sécurité Routière
CACES	Certificat d'Aptitude à la Conduite en Sécurité
CQP	Certificat de Qualification Professionnelle
CQP-ASA	Certificat de Qualification Professionnelle – Agent de Sureté Aéroportuaire
CQP-APS	Certificat de Qualification Professionnelle – Agent de Prévention et de Sécurité
DLL	Dynamic-Link Library
GUID	Global Unique IDentifier
ISO	International Organization for Standardization
SSIAP	Service de Sécurité Incendie et d'Assistance à Personnes
XML	eXtensible Markup Language
XSD	Xml Schema Definition

Concepts métier

Session : Cadre borné entre une date de début et une date de fin dans lequel des stagiaires suivent une formation. Elle peut être sanctionnée par un ou des examens.

Formateur : Personne en charge de la formation. Dans ce document, suivant le contexte, il sera parfois question de formateur, de testeur, d'enseignant ou bien d'animateur.

Stagiaire : Personne inscrite à une formation et/ou à un examen. En fonction du cadre d'utilisation du logiciel, on parlera de stagiaire, de participant ou bien d'élève.

Test : Nom générique regroupant à la fois les questionnaires projetés aux stagiaires dits tests théoriques mais aussi les épreuves pratiques.

1 Introduction

Next Media est à la fois une société de services informatiques (développement, maintenance d'applications informatiques) et une société de formation informatique certifiée Microsoft. Je travaille dans cette société comme développeur et formateur depuis Octobre 2005.

Je dispense ponctuellement, quelques jours par an, des formations sur des langages de programmation. Visual Basic et Visual Basic.Net sont les deux principaux langages sur lesquels je forme les stagiaires. J'anime aussi parfois des formations liées à l'utilisation de logiciels que nous développons. Ces formations m'apportent beaucoup, notamment au niveau du sens de l'écoute, et de la gestion des relations que l'on peut entretenir avec les clients.

Sur la partie développement, j'ai la charge de certains projets sur lesquels j'ai travaillé lors de mes débuts dans la société. Ces développements m'ont permis d'acquérir de l'expérience, notamment sur le développement objet et l'ergonomie. Ils sont aujourd'hui terminés et je ne gère dorénavant que la maintenance et les éventuelles évolutions que nous demandent les clients. Le projet principal sur lequel je travaille à 95% de mon temps et sur lequel ce mémoire s'articule est le projet QuizzBox.

2 Next Media

2.1 Son histoire

La société Next Media a été créée en 1988 par messieurs Jean-Marc Nermond et Antoine Pannetier. Elle est implantée dans le parc technologique de La Pardieu à Clermont-Ferrand et elle a comme activité la formation informatique. A ses débuts, l'entreprise dispose d'un effectif de 3 personnes.

En 1989, l'entreprise développe son activité de formation en embauchant 3 formateurs.

Puis en 1992, c'est au tour de la formation AutoCAD d'évoluer ; 1 formateur, 2 développeurs et 1 technicien AutoCAD sont embauchés.

En 1995, une scission au sein de l'entreprise fait de Jean-Marc Nermond l'unique gérant de Next Media.

En 1999, la partie développement d'applications de la société connaît un essor. Next Media déménage dans des locaux trois fois plus grands qui contiennent 7 salles de formation. De nouveaux formateurs et développeurs sont embauchés, ainsi qu'un commercial.

En 2000, l'entreprise Internet Evolution devient une filiale de Next Media. Spécialisée dans le développement de sites Internet, cette filiale sera rachetée en 2005 par un groupe de communication.

En 2001, c'est le lancement de la première version du système de vote interactif QuizzBox.

Depuis 2002, Next Media continue de développer toutes ses activités.

En 2004, elle est devenue le premier centre à pouvoir décerner sur la région Auvergne la certification Pearson Vue Authorized Center.

En 2005, Next Media est devenu le seul centre CPLS (Certified Partner Learning Solutions) de toute la région Auvergne-Limousin. Cet agrément lui a permis de former de nombreux techniciens sur les technologies Microsoft, aussi bien dans le développement d'applications que sur la mise en place d'infrastructures ou encore de la mise en place de bases de données.

C'est également en 2005 que QuizzBox est retenu par le Ministère de l'Intérieur, parmi trois systèmes, pour les examens de sécurité incendie SSIAP (Service de Sécurité Incendie et d'Assistance à Personnes).

Et depuis le 1er Septembre 2007, Next Media est devenu Prometric Testing Center. Avec l'agrément Pearson Vue, Next Media est un centre permettant de décerner des certifications sur toutes les technologies de l'information. Les formateurs certifiés Microsoft et leur réseau informatique ont permis la mise en place de Thomson Prometric.

Aujourd'hui ce sont plus de 3000 entreprises qui ont fait confiance au savoir-faire de l'équipe de formateurs et de développeurs de Next Media. Et ce sont plus de 1000 QuizzBox vendus, notamment pour une utilisation dans le domaine de la formation en sécurité (incendie en particulier).

2.2 Situation géographique

La société Next Média est située à Clermont-Ferrand, au cœur de l'Auvergne, dans le département du Puy-de-Dôme. Elle est implantée dans la zone technologique de La Pardieu.

2.3 Le personnel

L'entreprise emploie une quinzaine de salariés, hommes et femmes de tous âges.

Next Media ayant différentes activités, les employés sont répartis en 4 équipes :

- Les formateurs, qui travaillent pour la partie formation informatique de la société ;
- Une première équipe de développeurs qui se charge du développement d'applications RAD ;
- Une seconde équipe de développeurs qui s'occupe du développement et du suivi des systèmes QuizzBox ;
- L'équipe administrative et commerciale qui gère la partie vente et gestion.

2.4 L'organigramme

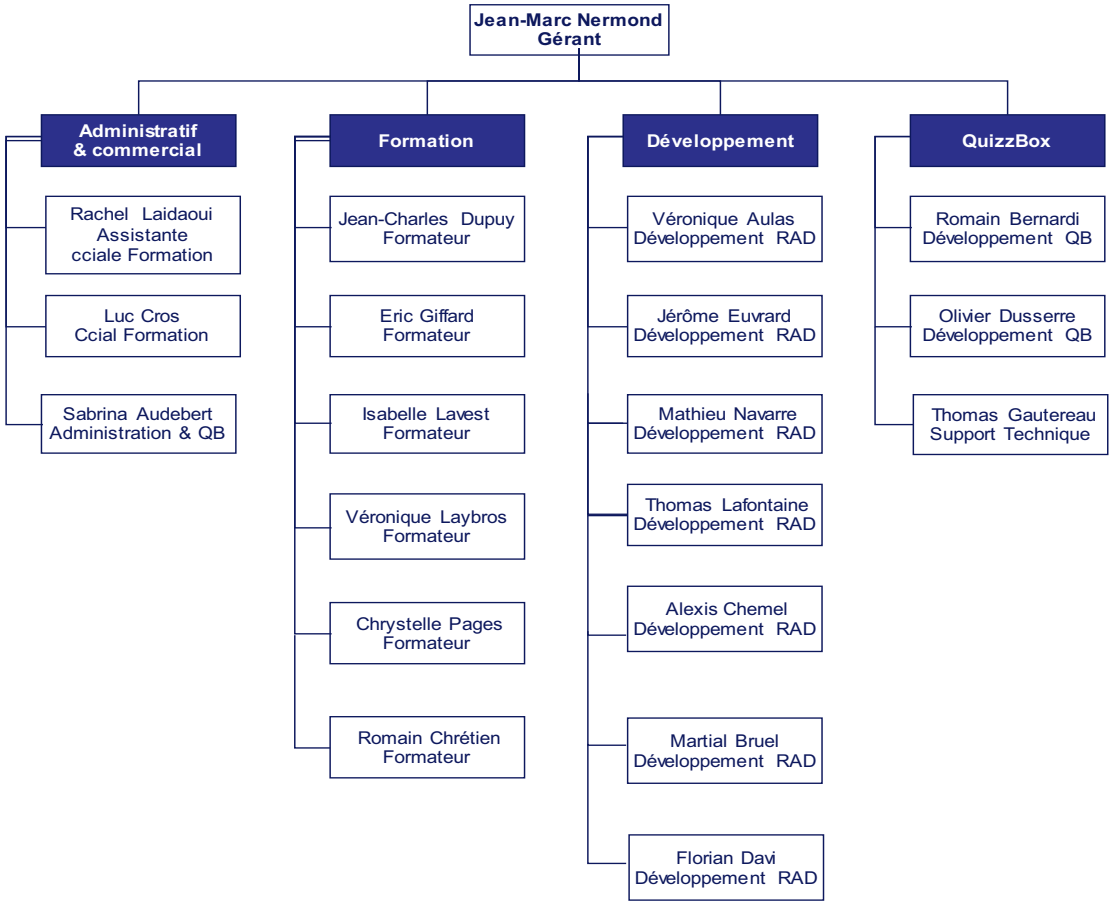


Fig. 1 – Organigramme Next Media

2.5 Présentation générale des activités de l'entreprise

Next Media est une société de service en ingénierie informatique. Elle propose à ses clients des formations informatiques et des développements d'applications.

2.5.1 Formation informatique

Les formations animées par la société sont basées sur différents domaines tels que la bureautique, internet, les bases de données, le développement, la gestion de projets, etc. Une formation est basée sur des cas pratiques et la réalisation de documents professionnels par les stagiaires. Les méthodes pédagogiques s'appuient sur les méthodes actives adaptées aux adultes. Les stagiaires sont invités à s'exprimer de façon à les impliquer et à créer un climat de groupe positif et interactif. Le formateur combine des exposés concis et des mises en situation. Écoute active et respect du rythme de chacun sont les priorités. Chaque stagiaire se verra remettre un support de formation qu'il pourra garder par la suite. Ces supports sont réalisés et actualisés en interne pas nos formateurs. Ce support reprend les concepts et modes opératoires vus au cours de la formation.

2.5.2 Développement d'applications

Next Media propose des solutions de développement informatique dans de nombreux domaines ; la technologie .Net avec des outils comme SharePoint, Visual Basic mais aussi Excel, Access, PHP-MySQL, etc.

Depuis quelques années maintenant, l'entreprise s'est lancée dans une nouvelle activité, le développement du système QuizzBox, un système de vote par télécommande qui rend les présentations PowerPoint interactives. Je présenterai ce système dans la partie suivante de ce rapport car il est le sujet du mémoire.

3 La solution QuizzBox

3.1 Qu'est-ce que QuizzBox

Conçu et distribué par Next Media depuis 2001, QuizzBox est un système de vote interactif. QuizzBox est utilisable avec tout dispositif TIC (tableau blanc interactif, écran vidéo, téléviseur, …). Il fonctionne à l'aide de boîtiers de vote et d'un récepteur USB.

Il s'appuie sur le logiciel Microsoft PowerPoint qui permet à l'aide d'une nouvelle barre d'outils, « la QuizzBar », de créer facilement des questionnaires. De plus, il s'allie facilement avec les fichiers PowerPoint déjà existants.

Depuis sa création, deux versions du matériel ont été commercialisées :

Fig. 2 – Système QB1

- Les systèmes QB1, composés de télécommandes sans afficheur utilisant un boîtier avec antenne comme récepteur. Ces systèmes ne sont aujourd'hui plus vendus. Nous proposons néanmoins à nos clients une offre de reprise de leur matériel.

- Les systèmes QB2, composés de télécommandes avec affichage digital, connectées à un récepteur USB. C'est le système que propose aujourd'hui la société. Il est composé de télécommandes (jusqu'à 60), d'une télécommande administrateur et d'un récepteur radio (RHF 433 Mhz) branché sur un port USB (format clé USB). Il est également vendu avec un CD d'installation et une notice d'utilisation, le tout étant emballé dans un sac à dos et une housse de transport.

Fig. 3 – Système QB2

Chaque télécommande, comporte 10 touches (8 touches de réponses, 1 touche de validation et 1 touche d'annulation), un écran qui permet l'affichage des réponses données, et un voyant de signalisation de batterie faible. Pour ce qui est de la télécommande

administrateur, les touches ont une fonctionnalité spécifique permettant à l'animateur de naviguer dans le questionnaire et d'afficher les résultats en instantané.

Ce système de vote est donc idéal dans un cadre commercial ou professionnel pour dynamiser une réunion et apporter de l'interactivité dans un séminaire de formation. Idéal également pour les présentations PowerPoint grâce à l'intégration directe de tout document existant ou nouvellement créé. Les animations, vidéos ou documents sonores, y trouveront leur place sans aucune difficulté. Dans le cadre de l'éducation, QuizzBox dynamise les cours, les rend plus ludiques et devient un véritable outil d'évaluation grâce à la correction automatique des tests et l'affichage instantané des résultats.

QuizzBox est utilisé dans différents domaines comme par exemple:

- habilitation incendie (SSIAP), habilitation électrique, habilitation conduite d'engins (CACES) ;
- animation d'ateliers culturels (musées, expositions) ;
- conduites de réunions commerciales ;
- évaluations, sondages et enquêtes de satisfaction.

3.2 Fonctionnement de QuizzBox

Le système est constitué d'un récepteur USB ainsi que de 15 à 60 télécommandes pour les stagiaires plus une télécommande pour l'administrateur, le tout fonctionne par radio fréquence. Les questionnaires sont créés à partir de fichiers Microsoft Excel ou présentations Microsoft Powerpoint. Une barre d'outils, un ActiveX en fait, permet de rendre les diapositives interactives. On peut créer soit des questionnaires de connaissance soit des questionnaires de satisfaction. QuizzBox est actuellement utilisé par plus de quatre cents clients soit pour faire de la formation soit pour faire passer des examens. Parmi ces clients, je peux citer Bureau Veritas, Socotec, Michelin, Claas Academy, une grande part des centres de formation agréés pour faire passer les examens de sécurité incendie et sécurité civile ainsi que beaucoup d'autres centres de formation. La mise en application du

programme est très simple. Après avoir créé un questionnaire, il suffit de saisir la liste des participants et de leur affecter un numéro de télécommande, puis de choisir son questionnaire et cliquer sur « Démarrer ». Les stagiaires se trouvent face à une présentation de type Powerpoint (généralement projetée via un vidéo projecteur) avec en haut, un bandeau qui affiche leur numéro de boîtier qui indique s'ils ont répondu à la question en cours. Les boîtiers sont munis de huit touches numérotées plus une touche de validation ainsi qu'une touche d'annulation.

D'un point de vue matériel, QuizzBox utilise des télécommandes sans fil à ondes radios hautes fréquences (RHF) et nécessite l'installation d'une version du pack Microsoft Office 2002 ou supérieur, sur un système d'exploitation Windows XP, Vista ou Seven.

Pour utiliser QuizzBox, on commence par l'installation du logiciel qui met à disposition dans les versions standards, trois modules :

- Le module utilisateur « QB_Utilisateur », qui permet à l'animateur de créer une session, de lancer un examen, de consulter les résultats à l'issue de l'examen ;
- Le module d'administration « QB_Administrateur », qui permet de paramétrer le système et de définir les sécurités. ;
- Le module de gestion des questionnaires « QB_Questionnaire », qui permet de gérer les questionnaires, et de paramétrer les questionnaires à tirage aléatoire.

QuizzBox s'intègre entièrement dans PowerPoint. Les questionnaires sont conçus directement dans ce dernier à l'aide d'une barre d'outils QuizzBox et sont diffusés sous PowerPoint.

Une fois cette installation terminée, aucun paramétrage n'est nécessaire pour le matériel. Le pilote du récepteur étant certifié Microsoft, il ne nécessite aucune installation (Plug & Play).

QuizzBox est prêt à être utilisé. La première étape consiste à réaliser son questionnaire à partir de PowerPoint et sa barre d'outils QuizzBox.

On ajoute une nouvelle diapositive. Dans la section « Titre » on inscrit la question que l'on veut poser et dans la section de texte on saisit les réponses en utilisant des puces numérotées pour que chaque réponse soit préfixée d'un numéro. Il suffit ensuite de rendre la diapositive interactive en sélectionnant la bonne réponse et en cliquant sur « Ajouter réponse juste » de la barre d'outils.

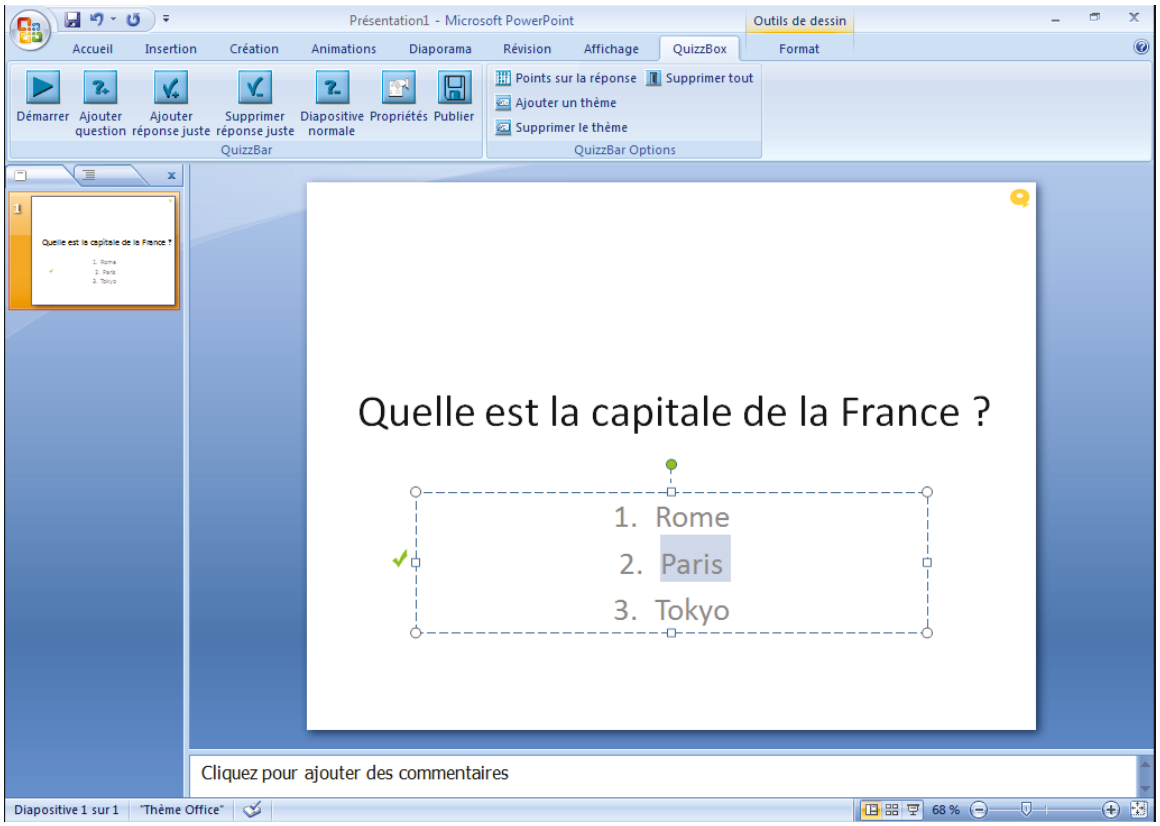


Fig. 4 - Exemple de question

On répète l'opération autant de fois qu'il y a de questions dans le questionnaire. Il ne reste alors plus qu'à publier le questionnaire avec le bouton *Publier*.

Il est également possible de créer un questionnaire à thèmes ou un questionnaire à tirage aléatoire, généré à partir de plusieurs sources PowerPoint depuis l'interface « QB_Questionnaire » (cf. Annexe 4).

Ensuite, il faut créer une session à partir de « QB_Utilisateur ». On lui donne un nom et on saisit la liste des participants auxquels on affecte un numéro de télécommande. Puis on choisit son questionnaire et on démarre en cliquant sur le bouton *Démarrer* à droite de la fenêtre.

Le questionnaire se déroule, les personnes répondent aux questions et à chacune d'elle les réponses sont envoyées instantanément dans la base de données QuizzBox. Ces résultats sont obtenus grâce à un système de comptage de points qui ramène à une note sur vingt.

Une fois le questionnaire terminé, on accède aux résultats dans QB_Utilisateur par l'onglet « Résultats ».

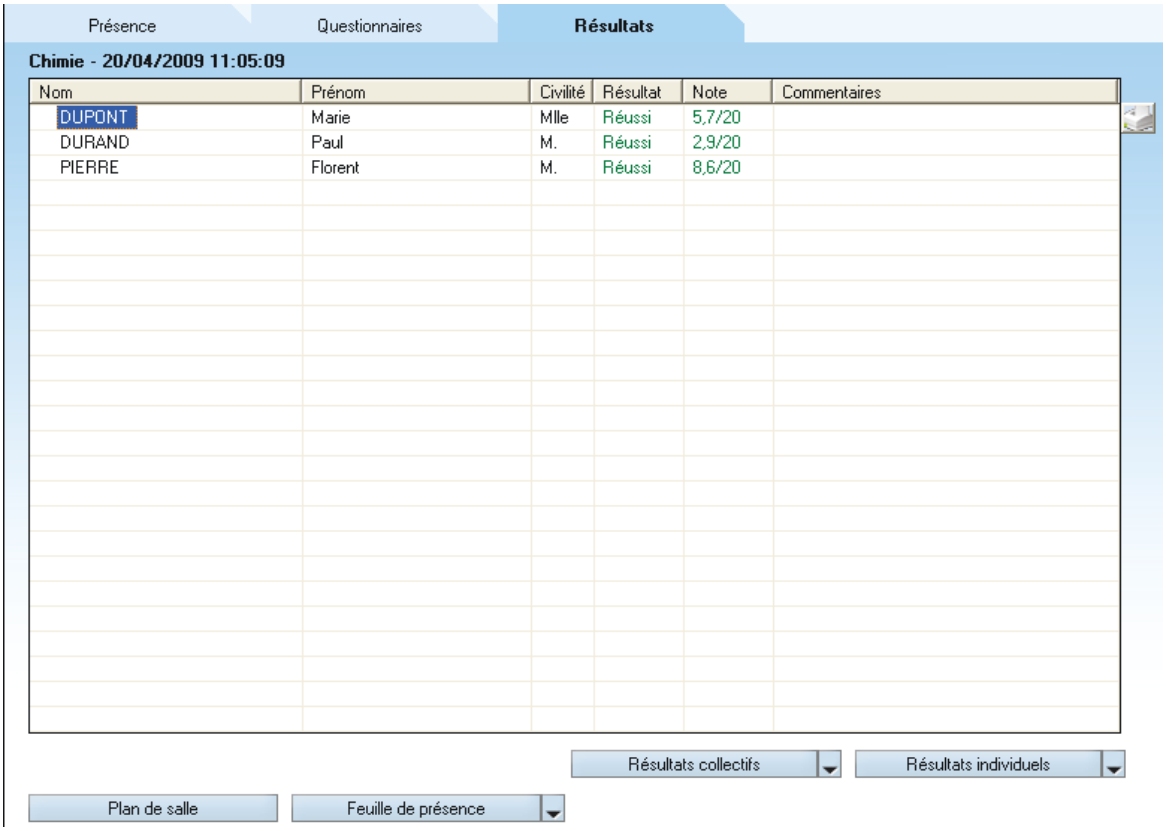


Fig. 5 - QB_Utilisateur, onglet "Résultats"

A partir de là, il est possible d'éditer un fichier de résultats individuels. Pour ce faire il suffit de cliquer sur le bouton « Résultats individuels ». Plusieurs modèles de résultats sont mis à disposition en cliquant sur la flèche à côté du bouton ou par clic-droit sur ce bouton.

Cliquer sur un des modèles de la liste entraîne l'édition des résultats individuels en tenant compte du nouveau modèle sélectionné.

Il est également possible d'afficher les résultats collectifs mais cette fois-ci avec le bouton « Résultats collectifs ». Tout comme pour les résultats individuels, ces résultats peuvent être édités selon différents modèles.

3.3 QuizzBox standard

Un QuizzBox standard est composé d'un sac à dos qui contient une mousse pouvant accueillir jusqu'à 60 télécommandes plus le récepteur ainsi que le CD d'installation et la documentation sous forme de livret. On parle de version standard car c'est la suite logicielle qui est fournie par défaut avec le matériel dans un cadre d'animation ou de formation classique. Nous disposons de plusieurs autres versions basées sur la version standard, dédiées à des clients spécifiques ou à des cas d'utilisation particuliers.

3.4 Les différents outils

La solution QuizzBox met en œuvre plusieurs outils déployés à différents niveaux.

On trouve d'une part les outils déployés en local sur les postes des formateurs dont le but est de faire de l'animation, de la formation, de l'évaluation. Dans la suite du mémoire, le terme QuizzBox fera référence à ces outils.

D'autre part, nous avons des outils dédiés à la consolidation de données, à l'envoi de résultats, à la gestion de contenus. Les transmissions de données peuvent se faire par LAN, FTP ou même par mail. Ces autres programmes font partie de QuizzBox Solution.

3.4.1 QuizzBox

Ci-dessous un schéma descriptif des outils installés sur les postes des formateurs.

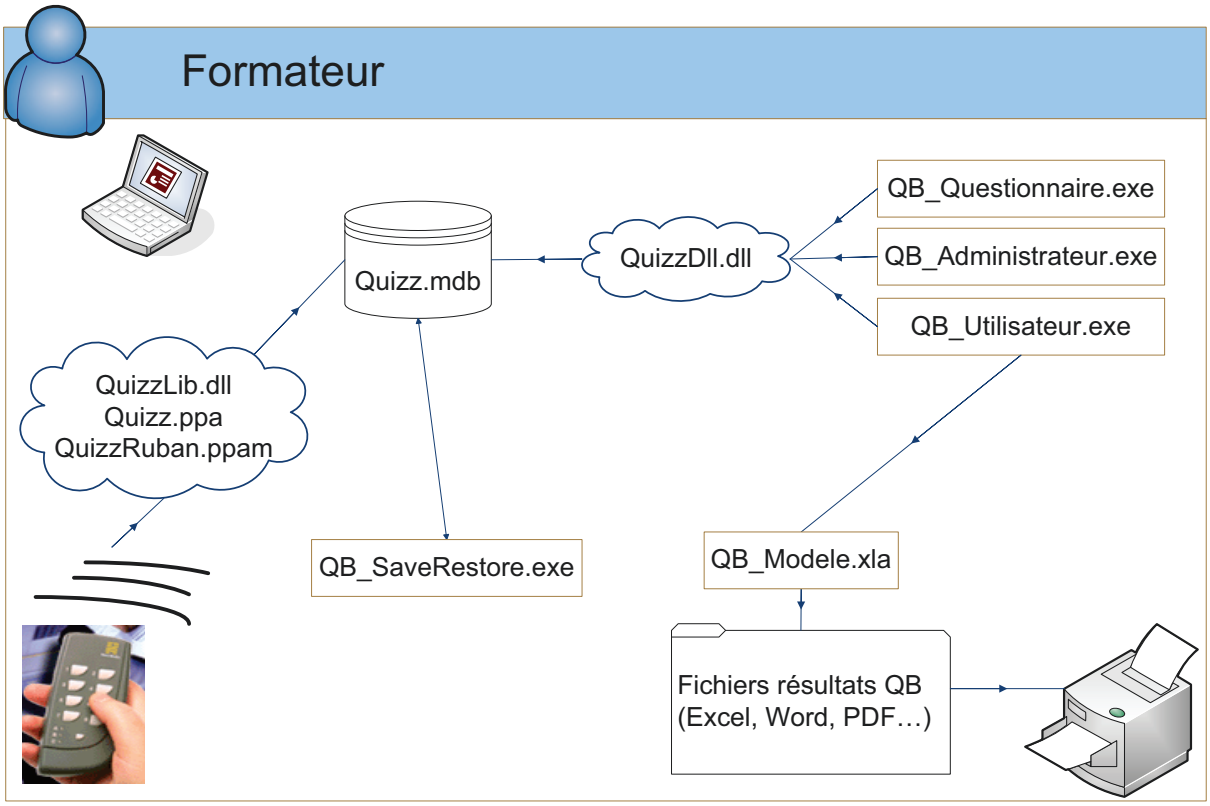


Fig. 6 – Schéma des différents programmes de QuizzBox

- QB_Utilisateur : Ce programme est déployé sur chaque poste formateur avec le master d'installation QuizzBox. Il permet de créer, d'animer et de consulter les résultats d'une session. C'est cet outil qui est principalement au centre de ce mémoire.

- QB_Administrateur : QB_Administrateur, comme son nom l'indique, est un module d'administration. Il permet de gérer les modèles utilisés pour générer les feuilles de présence, les résultats collectifs et individuels. Il permet également de modifier le nom de l'entreprise, l'adresse complète, le numéro de téléphone, le fax, la couleur dominante du modèle ainsi que logo (taille recommandée 123px * 79px). Enfin, QB_Administrateur permet de paramétrer certaines options de

QB_Utilisateur, de configurer la télécommande administrateur, de définir un récepteur multi-système, ainsi que de configurer un certain nombre d'autres paramètres.

- QB_Questionnaire : QB_Questionnaire est le programme qui permet de gérer les questionnaires publiés dans QuizzBox, ainsi que de créer des questionnaires multi-sources, des questionnaires avec thèmes et des questionnaires à tirage aléatoire.

- QB_SaveRestore : QB_SaveRestore est le programme de sauvegarde et de restauration des résultats des sessions passées avec QuizzBox. Actuellement utilisé lors de l'installation et de la désinstallation de QuizzBox, nous nous en servons aussi pour les imports et exports des données entre la base de données locale et des bases distantes utilisées par d'autres de nos outils comme QB_Consolidation par exemple.

- QB_Modele.xla : C'est une macro Excel dont la fonction est la restitution de résultats sous différents formats via de nombreux modèles. Elle prend en entrée des résultats sous la forme XML pour en sortie afficher des graphiques, tableaux Excel, Word ou PDF.

- QuizzLib.dll : C'est une librairie dédiée à la gestion du matériel. Elle est en charge de récupérer les réponses des stagiaires via leur télécommande. Elle a aussi pour fonction de gérer le récepteur et la télécommande administrateur.

- QuizzDll.dll : C'est le noyau de la solution, une librairie qui contient toutes les fonctions standards du QuizzBox et qui sont utilisées dans toutes les applications de la solution. Cette librairie évite la duplication de code dans tous les projets et donc en favorise leur maintenance.

- Quizz.ppa : Quizz.ppa est une macro Powerpoint utilisée pour gérer les évènements lors d'une animation utilisant QuizzBox. Il y a un lien étroit entre cette macro et QuizzLib.dll. La macro contient seulement le code qui effectue des

appels à QuizzLib.dll pour la gestion de l'animation. C'est aussi dans cette macro complémentaire qu'est définie la barre d'outils QuizzBox nécessaire à la conception d'un questionnaire, seulement pour les versions d'Office strictement inférieures à 2007.

- QuizzRuban.ppam : Utilisée seulement avec les versions d'Office supérieures ou égales à 2007, cette macro complémentaire créé un ruban dans Powerpoint remplaçant ainsi la barre d'outils des versions précédentes.
- Quizz.mdb : De type Microsoft Access, cette base de données stocke tous les résultats des tests passés avec QuizzBox, le référentiel des questionnaires enregistrés dans le système ainsi que les informations sur les sessions et leurs participants.

L'annexe 4 propose une capture d'écran de ces différents modules QuizzBox.

3.4.2 QuizzBox Solution

Nos clients possédant plusieurs systèmes QuizzBox ont tous peu à peu souhaité diffuser leurs questionnaires à tous leur formateurs, récupérer les évaluations des stagiaires, envoyer des contenus pédagogiques, etc… Pour répondre à ces besoins, nous avons développé plusieurs projets satellites autour de QuizzBox pour les formateurs. Ces logiciels font de QuizzBox une solution complète facilement intégrable dans un système d'information.

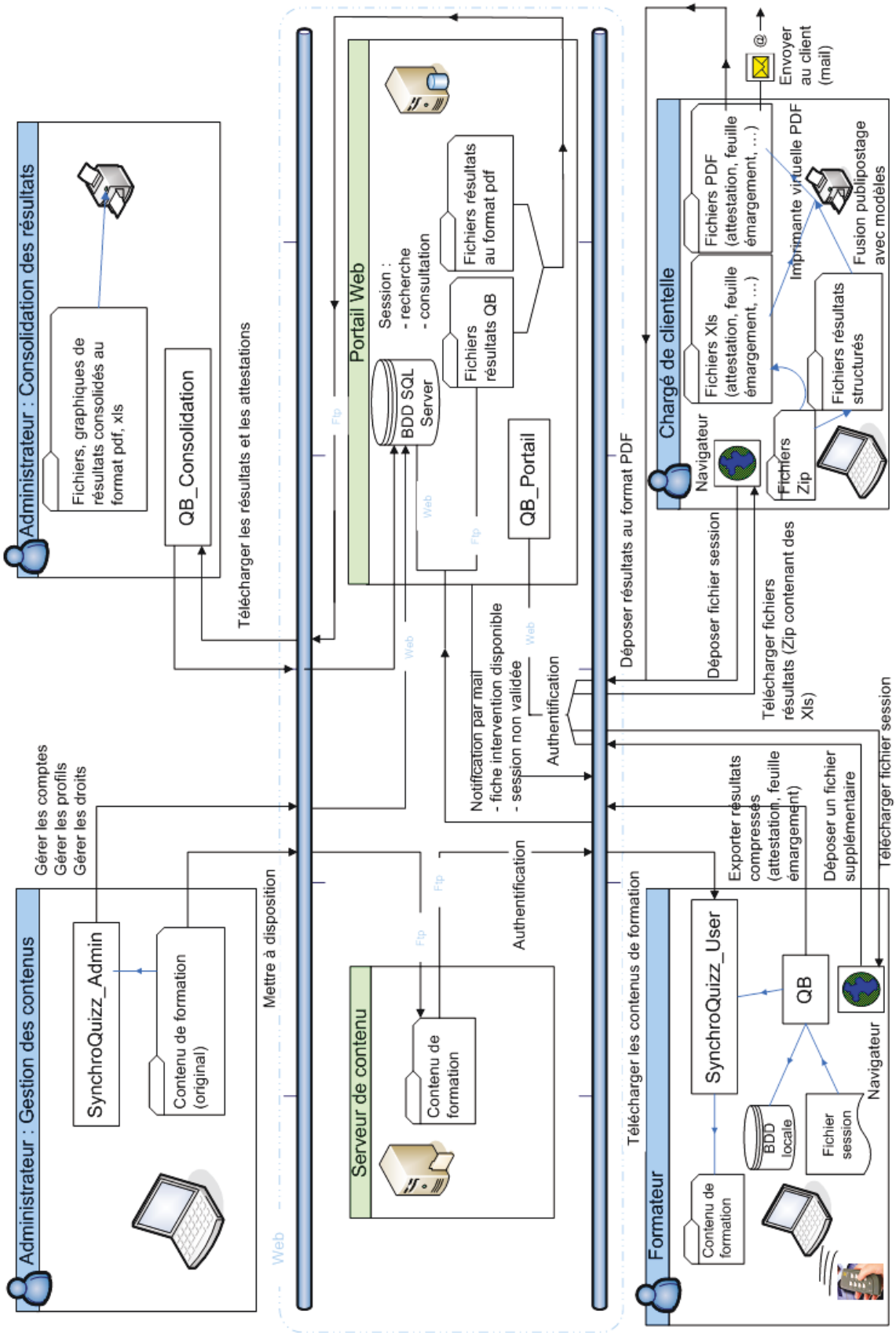


Fig. 7 – Les différents outils de la solution QuizzBox

- SynchroQuizz_User : Module externe mais néanmoins disponible via QB_Utilisateur, ce programme permet au formateur de récupérer et mettre à jour les contenus qui lui sont destinés, en fonction de son profil. Il est déployé avec le master d'installation QuizzBox. Quand une mise à jour des contenus est déployée sur le serveur, elle est directement disponible sur les postes des formateurs. J'ai développé cette application en Visual Basic .Net.

- SynchroQuizz_Admin : Cette application est installée sur les postes des administrateurs chargés de la mise à disposition des contenus (contenus accessibles pour les formateurs par SynchroQuizz_User). Elle permet de gérer les droits des utilisateurs (gestion par profil). Elle est déployée à l'aide d'un installeur. C'est un logiciel que j'ai développé en 2009 en .Net.

- QB_Portail : C'est une application que j'ai développé en PHP, installée sur un serveur Web. Elle sert d'interface entre les formateurs et les responsables de formation. Les formateurs se connectent via leur navigateur Web pour récupérer les fichiers des sessions qu'ils vont animer afin d'éviter les erreurs de saisie (société, nom des participants, …). Après utilisation de QuizzBox, en quelques clics, l'animateur rend disponibles les documents liés à la session (résultats, attestations, feuille d'émargement). Le chargé de clientèle ou le responsable de formation pourra les diffuser instantanément après vérification. Cette application permet également de centraliser toutes les sessions avec leurs résultats. En cas de contrôle ou d'audit, un document sur un stagiaire (attestation, détail d'un test) pourra facilement être retrouvé à partir de ce portail

- QB_Consolidation : Cette application a pour vocation de centraliser les résultats des tests et les évaluations sur les postes des formateurs et d'apporter des synthèses sur l'ensemble des données. Ces synthèses sont disponibles au travers de plusieurs modèles (tableaux, graphiques …) suivant une multitude de critères. La remontée des informations des formateurs peut se faire par LAN, FTP ou

même par QB_Portail. J'ai développé cet outil spécifiquement pour Claas Academy en 2009, il a depuis été vendu à d'autres clients.

- QB_Extranet : QB_Extranet est une application web que j'ai développé en ASP.Net, en utilisant le langage C#. Cet extranet a pour vocation de centraliser les résultats des stagiaires dans une seule base de données SQL Server et de pouvoir imprimer des rapports concernant par exemple l'évolution d'un stagiaire au cours du temps, l'évaluation moyenne d'un formateur par les stagiaires, les résultats collectifs ou individuels lors du passage d'un examen. Les résultats sont remontés depuis les interfaces QB_Utilisateur utilisées par les différents formateurs. A chaque ouverture de QB_Utilisateur, une synchronisation s'opère entre le système d'information de l'entreprise et la base de données locale du poste du formateur. Cette synchronisation est bidirectionnelle ; en effet, les résultats sont remontés mais les différents examens, sessions prévus sur une période de temps paramétrable, sont téléchargés vers chacun des formateurs. Toutes les communications se font via des fichiers XML, qui permettent de travailler en mode déconnecté, pour un envoi des fichiers par e-mail par exemple. Cet outil n'est plus proposé aujourd'hui car il est bien trop spécifique au client qui nous en a fait la demande. Néanmoins, QB_Portail, un des autres outils de la solution pourrait évoluer dans ce sens.

3.5 Les différentes versions

Aujourd'hui, il existe de nombreuses versions de QuizzBox. Suivant le cas d'utilisation, et en fonction du client, les CD d'installation ne sont pas les mêmes. D'un point de vue technique, ce sont les mêmes versions. On ne recompile pas les programmes pour générer ces différentes installations. Nous avons à notre disposition des scripts, qui suivant la version cible ajoute ou supprime des composants à l'installation, modifie des fichiers de configuration.

- Version 4 Standard : C'est l'installation type de QuizzBox. Composée des 3 programmes QB_Utilisateur, QB_Administrateur et QB_Questionnaire, cette

version est destinée aux clients faisant de l'animation, de la formation, de l'évènementiel.

- Version 4 Education : Identique à la version standard, cette version se distingue principalement par une modification des textes affichés dans les différents programmes pour s'adapter au monde de l'éducation. Par exemple, les concepts de « session », « stagiaire » et « formateur » sont remplacés par « classe », « élève » et « enseignant ».

- Version 4 SSIAP-CQP : Cette version est dédiée à l'entraînement et au passage des examens SSIAP et CQP (APS, ASA, ASC). Elle inclut un paramétrage spécifique pour que les utilisateurs soient positionnés par défaut sur ce type d'examens. Elle intègre aussi les pilotes d'une clé HASP-HL qui sert à déchiffrer les contenus de ces examens.

- Version 4 OPPBTP : Il s'agit d'une version dédiée exclusivement à la société OPPBTP dont l'activité est la prévention des risques dans le BTP. Par rapport à la version standard, cette version inclus le module de synchronisation SynchroQuizz_User. Ce programme permet aux différents utilisateurs, formateurs OPPBTP ou prestataires externes, de synchroniser les contenus, questionnaires mis à disposition par le responsable de formation depuis un FTP sur les postes en local. Une autre particularité de cette version est le chiffrement de tous les documents présents sur le FTP pour une utilisation exclusive avec cette version de QuizzBox. OPPBTP a aussi acheté nos autres modules de QuizzBox que sont QB_Portail, QB_Consolidation afin de centraliser et de pouvoir agréger les résultats des tests passés par les différents formateurs.

- Version 4 BV : Cette version de QuizzBox, à destination de notre client Bureau Veritas se distingue de la version standard par l'ajout dans l'installation des questionnaires réalisés par Bureau Veritas. Dans cette version, pas de module

de synchronisation. Les contenus ont été validés et ne sont pas amenés à être modifiés dans le temps.

- Version 4 Claas : Destinée à la société Claas Academy, dans un cadre de formation à l'utilisation de leur produit, cette version intègre elle aussi le module de synchronisation SynchroQuizz_User. Il se charge de mettre à jour en local les contenus disponibles sur un serveur de leur réseau mais aussi de faire remonter automatiquement les résultats des tests depuis les postes des formateurs vers notre outil de consolidation.

3.6 L'équipe QuizzBox et son fonctionnement

L'équipe QuizzBox est composée de 4 personnes.

Jean-Marc Nermond, le gérant de Next Media est aussi le commercial du produit. Fort de son expérience de formateur, il est également à l'origine du concept QuizzBox.

Olivier Dusserre embauché en 1997 comme stagiaire ingénieur pour développer une première version du produit est devenu le chef de projet QuizzBox.

Thomas Gautereau a rejoint l'équipe en 2010 et s'occupe du support téléphonique du produit, des formations techniques. Il est aussi en charge de la réception du matériel, de la préparation et de l'envoi des systèmes à nos clients.

Je suis pour ma part, développeur dans cette équipe. Mon implication dans ces projets est totale, et comprend les contacts avec les clients, l'établissement des cahiers des charges, l'analyse, les spécifications, le développement, les tests, les passages en production, la formation des utilisateurs, la maintenance corrective et évolutive du produit. Je travaille directement avec le chef de projet. Chacun de nous est autonome sur les parties dont il a la charge, mais nous prenons beaucoup de décisions en commun sur les orientations du projet en général et même sur des choix d'interface homme-machine, d'architecture... J'ai développé seul certains de ces programmes, notamment les logiciels de consolidation de résultats et de synchronisation de la solution QuizzBox. Nous utilisons pour nos

développements la solution d'archivage et de travail collaboratif de Microsoft appelée Visual Source Safe.

4 Nos secteurs d'activité, la concurrence

A ce jour, plus de 14000 tableaux blancs interactifs ont été vendus en France. Les boîtiers de vote, les équipements complémentaires, les ordinateurs, tablettes, etc. en sont la suite logique. Actuellement, nous vendons plus de 200 systèmes par an soit un chiffre d'affaires de 400 000 euros. Indépendamment des ventes, sur l'année 2009, les projets issus de QuizzBox ont généré un chiffre d'affaires de 50 000 euros.

Aujourd'hui nous sommes positionnés sur plusieurs marchés, et suivant les secteurs d'activité, les concurrents et les objectifs ne sont pas les mêmes.

4.1 Le secteur de la sécurité

J'ai été embauché en 2005 pour implémenter au sein de QuizzBox une solution permettant de faire passer des tests SSIAP. Le SSIAP est le terme générique donné aux examens certifiant des aptitudes en sécurité incendie et aux personnes. Il est en fait composé des 3 examens SSIAP1, SSIAP2 et SSIAP3. A cette époque, Eidiser et La Générale Multimédia se partageaient les ventes de systèmes de vote sur ce secteur d'activité. Quatre ans après, bien qu'il soit difficile de donner un nombre, la plupart des centres agréés à faire passer ces examens possèdent notre système. Eidiser, éditeur de la solution Télédis, s'est lui-même exclu du marché en vendant du matériel encombrant (deux valises d'une dizaine de kilos pour vingt télécommandes) et peu robuste. La Générale Multimédia concepteur du système Powervote se positionne elle plutôt sur le marché des grosses assemblées, plusieurs centaines de participants. Sur le marché de la sécurité, leur système est peu adapté.

Un décret impose depuis 2005 l'utilisation d'un système de vote interactif pour faire passer ce type d'examen. Depuis maintenant quelques années, c'est notre système qui est préconisé par le ministère l'intérieur quand un centre de formation obtient l'agrément permettant de former et faire passer des examens dans les domaines de la sécurité incendie et civile.

4.2 Le secteur de l'éducation

L'éducation est aujourd'hui pour QuizzBox assurément le marché avec le plus gros potentiel, mais c'est aussi le plus dur à aborder. Les cycles de décision sont très longs, et malgré une politique gouvernementale visant à combler notre retard sur les anglo-saxons sur l'école numérique, nous ne vendons que quelques systèmes par an aux collèges, lycées, CFA.

C'est la mise en place cette année des examens sur la sécurité routière (ASSR, AER, ASR) qui va peut-être accélérer l'arrivée des systèmes de vote dans les écoles. Nous sommes aujourd'hui 9 entreprises dont les solutions de vote interactif sont retenues pour faire passer ces examens. Nous ferons tout pour faire partie des 2 ou 3 sociétés qui s'imposeront naturellement au fur et à mesure que les établissements s'équiperont.

4.3 Les autres secteurs d'activité

4.3.1 Les formations interactives

Une session de formation interactive donne la possibilité d'identifier le niveau des connaissances ou l'état des représentations initiales, de contrôler l'assimilation des nouvelles informations ainsi que de les mettre immédiatement en application (études de cas). La remise d'un rapport de synthèse individuel à l'issue de la session peut compléter la formation.

4.3.2 Les tests automatisés

Dans ce cadre, QuizzBox permet de faire passer des tests ayant pour finalité une habilitation, une certification, une attestation, etc.

4.3.3 L'évaluation et la mesure

En début de formation, le formateur fait passer un test pour mesurer le niveau initial.

En fin de formation, il effectue une mesure du niveau de connaissances, il est ainsi à même de mesurer la progression.

En fin de formation, le formateur procède également à une évaluation à chaud de la satisfaction.

4.3.4 Les réunions interactives

Des réunions sont régulièrement organisées avec une télécommande par groupe de travail. Une question est posée, les participants discutent et argumentent avant de se mettre d'accord sur une réponse. Les résultats sont ensuite visibles par tous.

4.3.5 Les Serious Games, les séminaires de motivation, lancements de nouveaux produits, réunions de fin d'année

En l'espace de dix ans, le champ d'application du vote électronique interactif s'est considérablement élargi. Ce type d'événements se déroule fréquemment selon un double mode d'interactivité : une première partie anonyme (sujets plus "sensibles") et une seconde partie avec identification individuelle des participants (sujets plus "ludiques"). La remise d'un Award aux vainqueurs clôture utilement la partie ludique de l'événement.

La manifestation peut également prendre la forme d'une compétition par équipe. Dans ce cas, les participants de la même équipe sont regroupés, par exemple autour d'une table. Chaque équipe dispose d'un seul boîtier pour exprimer la préférence du groupe. Ce dernier scénario est particulièrement approprié pour stimuler les échanges et renforcer les liens entre les personnes.

4.3.6 Les animations interactives de stand

Cette application 100% originale est une parfaite illustration du concept d'interactivité intelligente sur lequel repose le système QuizzBox. Il permet notamment d'organiser un sondage professionnel ou une formation interactive sur un stand durant une exposition ou une foire.

4.3.7 Les études de marché

Le vote électronique interactif permet à chacun de s'exprimer individuellement, sans s'influencer, avant de se justifier. Ce sont des atouts appréciables dans l'animation d'ateliers de créativité comme dans la conduite de tables rondes destinées à tester de nouveaux concepts (communication, R&D) ou des projets publicitaires. L'animateur est seul à disposer d'un tableau des votes individuels à l'issue de chaque question. Cette information lui permet d'approfondir les motivations des choix individuels. Le système de vote interactif est utilisé à chaque étape importante de la réunion.

4.3.8 Les réunions d'assemblées - Conseils municipaux

Le vote de résolutions, la prise de décision et la genèse de consensus font partie des applications les plus évidentes du vote électronique. Les avantages sont flagrants : gain d'efficacité (plus grand nombre de résolutions votées), gain de temps (rapidité de récolte des votes), gain de personnel (informatisation du processus). La combinaison "boîtier de vote - carte à puce individuelle" + "logiciel approprié" permet d'effectuer un vote nominatif, un vote pondéré et/ou un vote certifié, en toute sécurité.

4.4 Nos points forts

4.4.1 Échanges et interactions

A la différence des questionnaires sur Internet, QuizzBox permet interactivité et émulation.

4.4.2 Gain de temps

Actuellement, les examens (théoriques et pratiques) sont faits sous la forme d'une interrogation sur papier : écrire, faire passer l'examen, le corriger. QuizzBox permet la saisie par l'intermédiaire des télécommandes et la correction automatique. Dans le domaine de

l'entreprise, les formateurs et consultants sont souvent payés pour la formation. L'évaluation et la certification leur posent problème.

4.4.3 Pédagogie interactive

Un enseignant fait passer des tests de 5 à 20 questions, la correction des copies est automatique, il doit simplement mettre une appréciation sur la copie. En début de cours, ou pour une révision sur un chapitre, notre mode « instantané » permet à l'enseignant d'interroger les élèves sans avoir à préparer de questionnaire au préalable.

4.4.4 Sollicitation de l'élève

Au-delà de l'aspect ludique, l'élève est sollicité. En outre, lors d'un contrôle à partir de QuizzBox, la correction automatique permettra à l'élève de rectifier instantanément ses erreurs. L'évaluation n'est plus simplement sanction et mesure, elle redevient formative. Les élèves participent, ils ne se contentent pas d'écouter l'enseignant.

4.5 Nos faiblesses

4.5.1 Le matériel

Le système a été pensé à l'origine pour une utilisation en salle de formation avec potentiellement une quarantaine d'utilisateurs. Aujourd'hui, QuizzBox gère 60 télécommandes maximum par système. Cette limite est due à la technologie utilisée qui peut engendrer des problèmes de collision des données si trop de trames en provenance des télécommandes sont émises en même temps vers le récepteur. Ce nombre est suffisant pour les secteurs d'activité principaux sur lesquels nous nous positionnons, mais bien en dessous de celui nécessaire pour pénétrer le marché des grosses assemblées (>500 télécommandes) qui nous sollicitent de plus en plus. Nous faisons actuellement faire une nouvelle étude de recherche et développement par la société Qomo basée aux Etats-Unis pour un système capable de gérer jusqu'à un millier de télécommandes.

4.5.2 Les moyens financiers

Notre essor est freiné par un manque de moyens financiers qui maintient notre démarche commerciale dans une situation relativement artisanale. Les a priori sur la situation géographique de notre entreprise (Auvergne) est un des facteurs au manque d'engouement des investisseurs. Il est vrai aussi que démarcher des investisseurs n'est pas facile et que nous n'avons probablement pas les compétences pour mener efficacement cette action.

5 Les nécessités d'une nouvelle version

Le décollage des ventes de QuizzBox s'est fait fin 2005 lors de la mise en place de la première version permettant de faire passer les tests en sécurité incendie (SSIAP). Le développement de cette version s'est fait en quelques semaines dans l'urgence. Il fallait répondre à un besoin pressant des centres de formation suite à un décret du ministère de l'intérieur imposant l'utilisation d'un système de vote interactif pour faire passer ce type d'examen.

Depuis plus de six ans nous travaillons sur le programme QB_Utilisateur de notre solution. Cet exécutable est le point d'entrée à toute utilisation de QuizzBox. Il s'est construit en même temps que nous apprenions à mieux comprendre le métier et les besoins de nos différents clients aux domaines d'activité si divers. Durant ces années, d'évolution en évolution, le simple programme des premières versions est devenu le produit central de QuizzBox.

Nous sommes arrivés à un moment critique du développement où notre application ne pourra pas supporter d'un point de vue structure, les modifications que nous voulons y apporter. Actuellement, chaque nouvelle fonctionnalité ajoutée est potentiellement source d'« effets de bord » et nous demande beaucoup d'énergie sur les tests. C'est pourquoi nous souhaitons redévelopper entièrement QB_Utilisateur. Cela doit devenir une application avec une architecture qui supportera non seulement les fonctionnalités des versions précédentes mais plusieurs autres nouveautés que nous souhaitons implémenter. De par sa position centrale dans la solution, la refonte de QB_Utilisateur aura des impacts sur tous les autres outils de la solution.

Les besoins que nous avons aujourd'hui sont en partie basés sur notre propre expérience du système et de son utilisation dans différents contextes, mais ce sont surtout nos clients qui sont le moteur du projet. Plusieurs d'entre eux, nous ont sollicité au fil des ans pour développer soit des modules de synchronisation de contenus, soit des outils de consolidation de résultats, etc… Nous sommes passés du master QuizzBox, c'est-à-dire les

programmes installés sur les postes des formateurs à une solution QuizzBox. La solution QuizzBox englobe d'autres outils qui font la passerelle entre les postes des formateurs et les systèmes d'information de leurs entreprises.

Bureau Veritas, société de service spécialiste dans la certification, la formation et le conseil, est un client QuizzBox qui possède à ce jour 110 de nos systèmes. Bien qu'ils possèdent un nombre conséquent de QuizzBox, ils ne disposaient pas de nos autres logiciels de la solution. Ils nous ont soumis des demandes d'évolution de notre solution qui allait dans le sens d'une refonte complète de la suite logicielle. Cette commande a été le déclencheur financier de ce projet dont nous aurions tôt ou tard lancé le développement. Ils souhaitent disposer d'une application informatique et pédagogique destinée au formateur, simple d'utilisation, sécurisée et pouvant communiquer avec les autres outils de gestion d'affaires et de stockage de fichiers informatiques de leur entreprise. Cette application doit permettre de communiquer plus efficacement et plus rapidement les informations dues à leurs clients (feuille d'émargement, attestations, …).

L'objectif du projet est donc double, répondre aux besoins de Bureau Veritas, mais aussi faire évoluer QuizzBox pour pouvoir faire bénéficier de cette nouvelle version à nos autres clients.

5.1 Les objectifs de Bureau Veritas

Les principaux objectifs de cette nouvelle version de QuizzBox pour ce client sont les suivants :

5.1.1 Améliorer la réactivité dans les échanges de documents

Notre solution devra être capable d'enregistrer électroniquement la signature des stagiaires sur les feuilles d'émargement. Ces dernières seront envoyées par mail quotidiennement au responsable projet de Bureau Veritas. Une telle signature sera aussi apposée au bas des feuilles d'évaluation ainsi que sur les attestations.

5.1.2 Ajouter de la sécurité et du contrôle dans l'utilisation de QuizzBox

Bureau Veritas ne souhaite pas que leurs contenus puissent être diffusés et utilisés avec d'autres QuizzBox que les leurs, mais aussi qu'il ne soit pas possible d'avoir accès à ces documents sans passer par QuizzBox.

5.1.3 Avoir un environnement spécifique CACES

Cette évolution souhaitée est fonctionnellement la nouveauté la plus ambitieuse demandée par notre client. Pour être admis à ce type d'examen, les candidats doivent réussir des épreuves pratiques et théoriques. Les épreuves théoriques sont déjà gérées dans QuizzBox, par le biais des questionnaires classiques QuizzBox, mais les tests pratiques ne le sont pas. Cet outil sera dédié aux testeurs sur le terrain en charge d'évaluer les stagiaires en conditions réelles.

5.1.4 Limiter l'utilisation de l'application suivant le profil

Les utilisateurs QuizzBox au sein de cette société ont des profils multiples. Il y a les formateurs, les animateurs, mais il y a aussi des formateurs externes à la société qui sont susceptibles d'utiliser notre produit. Il faudra que chaque profil n'ait accès qu'à ses propres contenus et qu'aux fonctionnalités qui entrent dans leurs attributions. Cette évolution ne sera pas abordée dans ce mémoire car elle entre dans la phase 2 du cahier des charges de la refonte de notre solution qui n'est pas traitée dans ce document.

5.1.5 Communiquer avec les outils de gestion d'affaires Bureau Veritas

Evolution souhaitée aussi en phase 2, il y aura des modifications à apporter à plusieurs logiciels de la solution pour une meilleure intégration de QuizzBox au sein de leur système d'information. Les différents outils concernés sont :

- QB_Portail pour la gestion des documents des formateurs post-session ;

- QB_Consolidation qui centralise ces documents et qui au travers du reporting offre une vision macro des formations à son responsable ;
- SynchroQuizz_Admin et SynchroQuizz_User qui se chargent de gérer la mise à jour des contenus mis à disposition des formateurs sur leurs ordinateurs.

5.2 Nos propres objectifs

Nous avons profité de l'opportunité donnée par notre client pour nous aussi établir un cahier des charges et améliorer de notre point de vue QB_Utilisateur. Ces évolutions étaient prévues de longue date, et la commande de Bureau Veritas a permis d'amorcer le processus.

5.2.1 Repenser l'architecture

La première version de QB_Utilisateur a été écrite en 2005 en Visual Basic 6. Les possibilités de structurer une application sont assez minimes. Au fur et à mesure du temps, l'application est devenue difficile à maintenir.

C'est un produit en perpétuel évolution avec des mises à jour régulières du noyau, mais aussi des demandes spécifiques. La version 4 de la suite logicielle QuizzBox n'aurait pas pu absorber toutes les nouvelles fonctionnalités.

De plus, nous avons développé il y a quelques années, une librairie (DLL) en langage .Net qui s'appelle QuizzDll.dll. Elle contient des fonctions qu'utilisent l'ensemble des produits de la solution QuizzBox. La version 4 de QB_Utilisateur développée en Visual Basic 6 ne peut accéder directement à ce type de librairies créées en .Net, elle ne peut utiliser que des librairies de type COM. Via le concept d'interopérabilité, nous avions à notre disposition une autre dll compilée en COM Interop (QuizzDllCom.dll) utilisée seulement par QB_Utilisateur. Cette deuxième dll ne contient qu'une liste de fonctions qui pointent vers celles de la dll .Net. Le problème est d'une part qu'il faut maintenir ces 2 dll parallèlement et d'autre part que les fonctions de la dll COM ne peuvent recevoir/retourner que des paramètres simples. Impossible donc d'utiliser les types complexes comme les listes triées,

les collections et autres types inconnus de Visual Basic 6 qui s'avèrent indispensables pour une programmation fortement typée.

Il est aussi nécessaire de repenser entièrement l'architecture de QB_Utilisateur avec la vision d'aujourd'hui.

5.2.2 Gestion de l'unicité des personnes

Depuis les premières versions de QuizzBox, une personne inscrite à plusieurs sessions est vue d'un point de vue système comme des personnes différentes. Jusqu'à présent, on ne pouvait pas afficher de résultats pour un stagiaire transversalement sur plusieurs sessions. Cette agrégation de résultats se fait seulement dans les outils de consolidation. QuizzBox devra dans cette nouvelle version proposer un suivi des participants.

5.2.3 Gestions des examens

L'ancienne version de QuizzBox permet de faire passer des tests qui sont tous indépendants les uns des autres. On peut cependant sur une même session faire passer plusieurs fois le même test et éditer les résultats via des modèles spécifiques de type avant/après. La version 5 se doit de proposer une gestion efficace des examens. On entend par examen, une suite de tests théoriques ou pratiques au sein d'une même session. C'est l'agrégation des résultats de ces tests qui définit la réussite d'un stagiaire à un examen.

5.2.4 Gestion des épreuves pratiques

QuizzBox permet de faire passer des épreuves théoriques. Des questions sont posées aux stagiaires via des présentations Powerpoint auxquelles ils répondent grâce aux télécommandes. Cependant, nombre d'examens dans les domaines d'activité de nos clients, pour être validés, nécessitent en plus de passer des épreuves pratiques. Les stagiaires vont sur le terrain avec un examinateur qui valide sur de nombreux critères l'aptitude des participants dans des cas concrets. Dans notre évolution de QuizzBox vers la notion plus globale d'examen, il est nécessaire de pouvoir prendre en compte des résultats à des

épreuves pratiques. Ces épreuves pourront être construites dans un nouveau module distinct de QB_Utilisateur. En effet, la plateforme cible de cette nouvelle application pourra être un Tablet PC, voire un Smartphone. Cette gestion des épreuves pratiques est aussi dans le cahier des charges de Bureau Veritas. Leur volonté, est de pouvoir faire passer les certifications CACES et les habilitations électriques.

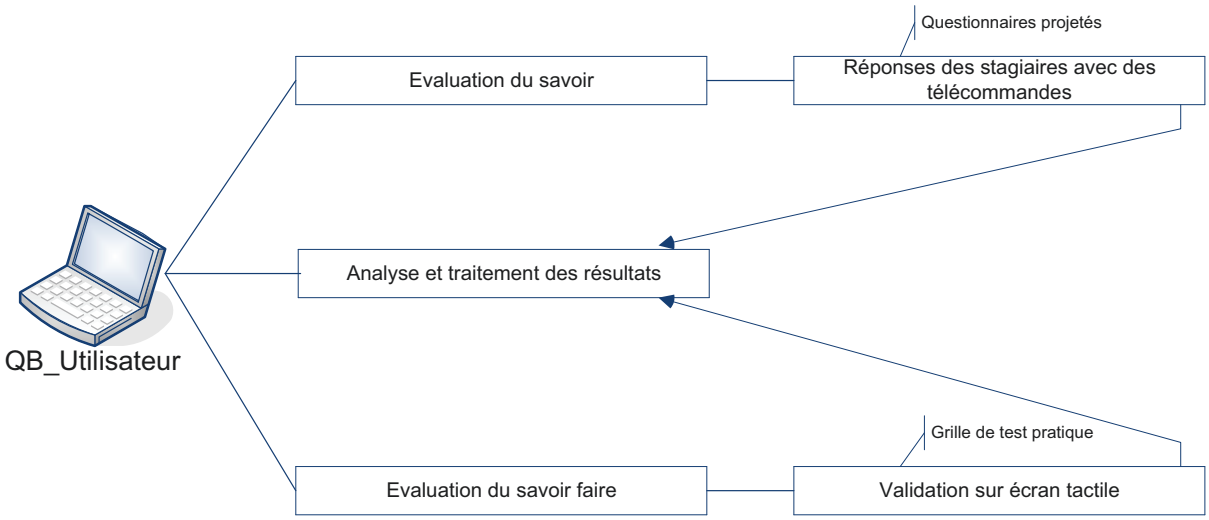


Fig. 8 – Schéma de principe de QuizzBox

5.2.5 Récepteur multi-applications

La gestion du récepteur dans QuizzBox est mono application. Une seule application peut accéder à un instant donné au récepteur, et donc à la réception de données depuis les télécommandes. Nous souhaiterions que le récepteur puisse communiquer avec tous les modules de QuizzBox en même temps et que la télécommande administrateur soit vraiment au centre de l'animation.

5.3 Critique de l'architecture de QB_Utilisateur version 4

Dans l'ancienne version, on ne pouvait pas vraiment parler d'architecture, car le langage, Visual Basic 6, dont nous disposions n'était pas adapté à de la programmation objet. Les notions d'héritage et de polymorphisme par exemple n'existent pas. La partie

métier de QuizzBox était seulement disponible via la base de données, ce qui est déjà structurant mais n'est pas satisfaisant.

Les évolutions que nous avons apportées en version 4 depuis ces derniers mois sont devenues source de problème. Cette absence de cadre dans le code a engendré de nombreux effets de bord. L'implémentation de toute nouvelle fonctionnalité prend beaucoup trop de temps.

C'est vraiment la complexité de la mise en œuvre de nouvelles évolutions sur la version 4 qui nous a fait ressentir le besoin d'avoir une application mieux structurée. Aujourd'hui, il nous faut repenser l'ensemble avec l'expérience acquise.

6 Conception de la nouvelle version, QuizzBox v5

La phase 1 du projet pour Bureau Veritas n'inclut que la modification des outils déployés sur les postes des formateurs, c'est ce qu'on appelle le master QuizzBox. Ce n'est qu'à la phase suivante, courant mai 2012, que les autres programmes de la solution seront modifiés, pour une intégration totale de QuizzBox au sein de leur système d'information.

6.1 La sphère métier

6.1.1 Cas d'utilisation

La version 4 de QuizzBox était déjà structurée autour de nombreux concepts métier. Ces concepts que sont les sessions, les questionnaires, les participants, ou les résultats par exemple, sont les mêmes à implémenter en version 5. Néanmoins, il faudra améliorer QB_Utilisateur et prendre en compte les nouvelles notions d'examens ou d'épreuves pratiques qui n'existaient pas dans les versions précédentes.

6.1.2 Problématique

Dans quelle mesure la refonte de QuizzBox doit-elle impacter la base de données Microsoft Access Quizz.mdb locale des formateurs ? En effet, nous nous devons de proposer à nos clients la possibilité de passer de la version 4 à la 5 sans perte de données, comme nous le faisions jusqu'à présent entre chaque mise à jour. Notre application QB_SaveRestore a la tâche de sauvegarder, à la désinstallation de QuizzBox, la base de données, ainsi que les différents fichiers ayant trait aux sessions et à leurs résultats. A l'installation d'une nouvelle version sur le poste, QB_SaveRestore est lancé et restaure les éventuels fichiers et la base de données. La première ouverture de QB_Utilisateur compare la version de la base restaurée avec la version en cours puis le cas échéant met à jour Quizz.mdb par un système de patches successifs. Si nous choisissons comme pour QB_Utilisateur de repartir d'une feuille blanche pour la conception de la base de données, la structure de Quizz.mdb pourrait être

totalement modifiée et nous devrons mettre en place un système très lourd pour passer la base de données de l'ancienne version à la nouvelle.

6.1.3 Réflexion et solution

La refonte complète ou modification en profondeur de la base de données Quizz.mdb a un trop fort impact. Il faudra dans un premier temps créer la routine de mise à jour de la base dans QB_Utilisateur / QB_SaveRestore qui sera très complexe. Nous devrons ensuite modifier les autres outils de la solution. En effet, nombre de nos outils utilisent la base de données locale pour des échanges de résultats par exemple. La solution que je préconise est de limiter les impacts sur la base de données.

6.1.3.1 Session active, liste de personnes

QB_Utilisateur doit être capable de gérer plusieurs sessions. On peut imaginer un formateur qui sur une semaine aurait un groupe de stagiaires les matins pour de la formation SSIAP, et un autre groupe les après-midi pour de la formation CQP-APS. Le formateur doit pouvoir basculer d'un groupe à l'autre dans l'interface. Néanmoins, à un instant donné dans QB_Utilisateur, il n'existe qu'une seule session active, celle sur laquelle on saisit des participants à un examen, où on regarde les résultats d'un test.

Il n'y aura pas de classe spécifique pour les sessions, mais une classe qui représente la session active (*SessionActive*), cf. Annexe 5. Si on veut basculer d'une session à une autre dans QB_Utilisateur, via la recherche de session ou la création d'une nouvelle session par exemple, la nouvelle session devient la session active. Dans QB_Utilisateur, on retrouve un certain nombre de propriétés de cette classe liées directement aux composants de l'interface ; zones de texte, listes déroulantes, cases à cocher… Toute modification de valeur de ces composants entraîne la mise à jour automatique de l'objet *SessionActive* instancié, puis en cascade de la base de données. Il en est de même dans l'autre sens, le constructeur de la classe *SessionActive* lit les données de la base de données Quizz.mdb pour créer le nouvel objet. De manière automatique, via des classes d'IHM, dont je parlerai dans la

gestion des interfaces, les informations de la session active sont affichées directement dans les composants dédiés à l'affichage des informations de session.

De la même manière pour les participants, au lancement de QB_Utilisateur, il importe peu de charger en mémoire l'ensemble des participants de toutes les sessions. Ce qui est important c'est de connaître et de pouvoir manipuler les participants de la session active. Cela donne lieu à une nouvelle classe intitulée *ListePersonneSessionActive*, qu'on retrouve en propriété de la classe *SessionActive*. En prévision de l'avenir, et parce que ce n'est pas encore implémenté pour le moment, j'ai aussi ajouté une classe *ListeEquipe*, qui permet de regrouper des participants avec chacun leur télécommande au sein d'une même équipe, ou bien de regrouper les participants dans une équipe avec une télécommande unique.

La classe *ListePersonneSessionActive* est ce qu'on appelle en .Net une *ListOF* de classes *PersonneSessionActive*. Ce type d'objet nous permet d'utiliser des méthodes de tri, de recherche, de parcours dans la liste typée d'une manière plus efficace qu'avec les types primitifs que sont les tableaux ou les collections. Pour obtenir ce fonctionnement, la classe *ListePersonneSessionActive* implémente l'interface *IList* et représente une collection d'objets non-génériques accessibles par index.

6.1.3.2 Gestion des tests, des contenus

Depuis l'interface de QB_Utilisateur, il est possible de lancer pour le groupe de personnes actif des tests théoriques, des tests pratiques et des contenus pédagogiques dans le cadre d'un examen. Il est aussi possible de lancer des tests théoriques ou des contenus hors examen. La gestion des épreuves pratiques ne s'effectue par contre que dans le cadre d'un examen.

Tous ces tests sont regroupés dans une classe abstraite *IHM_Test* qui contient les propriétés et méthodes communes aux différents tests. Chaque type de test hérite de cette classe abstraite et propose seulement les propriétés et méthodes qui lui sont propres, ainsi que les surcharges des méthodes de la classe abstraite. On retrouve ces classes en annexe 6 de ce mémoire. J'ai réparti les tests en plusieurs catégories :

- *IHM_TestClassique* : Questionnaires Powerpoint contenant des questions. Aujourd'hui il est encore question seulement de questionnaires Powerpoint, mais le développement de QuizzLib.dll fait par Olivier Dusserre fait déjà abstraction de cette technologie pour nous permettre un jour d'utiliser des vidéos, des images, etc…comme support au questionnement. Cette classe pourra elle aussi être dérivée en autant de classes que de technologies utilisées comme support.

- *IHM_TestPratique* : Questionnaires XML destinés à être utilisés par l'outil QB_Pratique.

- *IHM_TestMultiSources* : Questionnaires XML construits à partir de plusieurs sources Powerpoint, dans lequel on peut faire du tirage aléatoire de questions par exemple. La création de ces tests se fait dans l'outil QB_Questionnaire.

- *IHM_Contenu* : il ne s'agit plus de questionnaires mais de présentations Powerpoint servant à proposer par exemple du contenu pédagogique. Même si Bureau Veritas n'en a pas exprimé le besoin, un contenu dans QuizzBox est vu comme un fichier qui n'est pas un questionnaire. Il est donc possible d'avoir des contenus de type Excel, Word ou PDF par exemple.

- *IHM_XXX* : XXX correspond pour l'instant à SSIAP1, SSIAP2, SSIAP3. Ce sont des tests fortement typés qui obéissent à des règles particulières décrites directement dans la classe. Le SSIAP1, par exemple, est un test construit à partir de plusieurs sources Excel chiffrées. Les méthodes communes de la classe mère *IHM_Test* que sont *Preparer* et *Demarrer*, pour respectivement construire le questionnaire et le lancer via notre outil sont surchargées afin d'utiliser nos différents modules de chiffrement/déchiffrement ainsi que les fonctions permettant la sélection de questions dans des feuilles de calcul Excel.

La version précédente de QuizzBox est limitée à la seule gestion de tests théoriques et de contenus hors examen. La publication d'un test ou d'un contenu dans QuizzBox donne lieu à la création d'enregistrements dans les tables *tQuestionnaire*, *tModule*, *tQuestion* et *tReponse*. Ces tables correspondent à la sauvegarde des informations du questionnaire, de ses thèmes (modules), de ses questions et de ses propositions de réponse. Comme ces tests sont susceptibles d'être modifiés au cours du temps, nous avions choisi, quand un test est lancé sur une session et que des résultats commencent à être générés, d'enregistrer les éléments de ces tests dans des tables d'historique (*tSessionHistorique*, *tResultatHistorique* et *tReponseHistorique*). Ces différentes tables permettent d'éditer des résultats sur les tests passés même si le questionnaire d'origine a été modifié ou n'existe plus. La structure de ces tables n'est pas une structure relationnelle, mais une structure « à plat », qui se prête mieux à l'utilisation de nos différents programmes de restitution des résultats. J'ai choisi de garder cet existant pour limiter les impacts, sur la reprise des données entre les versions 4 et 5.

Dans notre base de données locale, la gestion des examens et des tests pratiques donne lieu à la création de nouvelles tables (cf. Annexe 12) qui sont *tExamen*, pour l'enregistrement des examens créés ; *tResultatExamen* qui stocke le résultat des participants aux examens et trois nouvelles tables d'historique que sont *tDomainePratiqueHistorique*, *tDomainePratiqueReponseHistorique* et *tDomainePratiqueResUserHistorique.* Ces tables reprennent la même structure que les tables créées pour enregistrer le détail des résultats à des tests théoriques. Il sera plus facile d'éditer des résultats si les tables de résultats des deux types de tests ont la même structure.

6.2 Les interfaces

6.2.1 Cas d'utilisation

Aujourd'hui QuizzBox est utilisé dans de nombreux domaines comme la formation en sécurité incendie, en sécurité civile, le passage et la délivrance de diplômes en habilitation électrique, en conducteur d'engins de chantiers. Il est aussi utilisé pour faire de l'animation évènementielle, par les professeurs dans les écoles dans la cadre de l'ASSR ou du contrôle

continu des élèves. Ces différents utilisateurs n'ont pas tous les mêmes envies, besoins, ni le même langage face au produit. Là où les formateurs en entreprise parlent de stagiaire, les professeurs dans les écoles parlent d'élève. La notion de société d'appartenance des stagiaires dans un centre de formation est importante, alors que dans l'éducation nationale, elle est inexistante. L'interface de QuizzBox doit s'adapter à chaque cas d'utilisation.

Au niveau de l'équipe QuizzBox, nous avons repris le mot interface pour définir le contexte dans lequel QB_Utilisateur est utilisé. Il existe autant d'interfaces que de cas d'utilisation. Chaque interface propose aussi des examens qui lui sont associés. Par exemple, une interface SSIAP, doit mettre à disposition trois examens (SSIAP1, SSIAP2 et SSIAP 3). Chacun de ces examens est composé de deux tests, un test théorique et un test pratique. La personnalisation de l'interface doit permettre d'adapter au cas d'utilisation les champs disponibles au niveau de la session, les informations affichées au niveau des participants. Il doit même être possible de choisir quels onglets pourront être visibles et quels seront les examens disponibles. Nous avons poussé la caractérisation de l'interface au thème à utiliser ainsi qu'aux menus qui seront adaptés aux besoins. La capture d'écran de QB_Utilisateur version 5 de l'annexe 4 désigne bien les différentes zones qui doivent être personnalisables.

6.2.2 Problématique

En version 4, pour pallier le problème, nous avions autant de versions de QuizzBox que de besoins différents. Il existait une version standard professionnelle, une version SSIAP / CQP-APS, une version éducation et d'autres versions spécifiques à des clients particuliers pour qui le QuizzBox est au centre du système d'information.

Les évolutions à apporter doivent permettre :

- De limiter le nombre de versions de l'application ;
- De personnaliser les informations de la session à afficher et le texte associé dans l'interface ;

- De personnaliser les informations des participants à afficher et le texte associé dans l'interface ;
- D'avoir une interface complètement paramétrable en termes de couleurs, images, police de caractère...

6.2.3 Réflexion et solution

L'utilisateur, après avoir installé l'application, doit se retrouver dans un contexte familier. Il serait curieux pour un professeur de lycée de lancer QB_Utilisateur et d'arriver par défaut sur une interface permettant de saisir des informations de société sur des stagiaires. Le point d'entrée pour l'utilisateur est donc l'installation. C'est pourquoi dorénavant, c'est le numéro de licence saisi à l'installation qui définit l'interface par défaut à utiliser.

Néanmoins, QuizzBox, dans le cadre des examens CQP-APS et CQP-ASA nécessite une clé physique de déchiffrement des questionnaires et de ce fait déploie lors de son installation les pilotes de ce matériel. Il ne serait pas très satisfaisant d'installer des pilotes pour un matériel qu'on n'utiliserait pas si on ne fait pas ce type d'examen. De plus, certains de nos clients, ceux qui possèdent plusieurs outils de la solution QuizzBox, disposent historiquement de leur propre version de notre système. En effet, l'installation de QuizzBox contient d'une part le paramétrage nécessaire pour la liaison avec les autres outils de la solution ; des comptes FTP, des comptes SQL Server, des mots de passe divers. D'autre part, l'installation dans certains cas, déploie aussi des contenus de formation évitant dans un premier temps aux formateurs d'avoir à synchroniser sur leur poste trop de données. Dans certaines sociétés, les contenus atteignent facilement plusieurs giga-octets. Là encore, nous proposerons encore des versions spécifiques à nos clients.

La multiplication des versions n'augmente que sensiblement notre charge de travail. Les programmes sont les mêmes pour tous les clients. Il n'y a pas ce qu'on appelle techniquement de « fork » dans le développement des logiciels ; à savoir une rupture dans le projet visant à avoir deux versions différentes basées sur le même noyau mais évoluant

distinctement. Seuls les fichiers de configuration et les contenus pédagogiques changent. Nous nous servons de l'outil d'installation professionnel InstallShield, pour générer ces différentes versions au travers de scripts.

Toutes les informations liées à la personnalisation de l'application doivent être stockées quelque part sur le disque de l'utilisateur. L'enregistrement en base de données aurait été possible sachant que QuizzBox utilise déjà une base de données Microsoft Access pour stocker les informations sur les personnes, les sessions, les résultats, etc… Cependant nous avons choisi le format XML pour plusieurs raisons. XML est un standard, les fichiers de ce type peuvent être lus par la majorité des plateformes ce qui en fait un format d'échange idéal. De plus, ils sont facilement compréhensibles et modifiables même par un public non informaticien.

L'interface de QB_Utilisateur est désormais décrite dans des fichiers XML sur deux niveaux. Par exemple, pour une interface de type SSIAP, l'application charge en premier les informations du fichier *standard-pro.xml* qui est l'interface standard professionnelle puis celles du fichier *ssiap.xml*, qui complètent ou supplantent celles du fichier standard. Un exemple de ce fichier de description est détaillé un peu plus loin dans ce chapitre.

Le programme QB_Utilisateur doit pouvoir charger aisément les informations contenues dans les fichiers XML d'interface. En effet, il est possible depuis l'IHM de basculer d'une interface à une autre. Il faut donc pouvoir stocker ce paramétrage d'interface dans une structure objet. Il aurait été fastidieux de recourir à de la lecture XML depuis Visual Basic .Net pour la création de ces objets, même si aujourd'hui les méthodes de lecture via des requêtes XPath facilitent le travail. C'est pourquoi j'ai choisi un principe en deux étapes. La première étape consiste à dé-sérialiser les fichiers XML. La dé-sérialisation est un mécanisme facile à mettre en place en .Net, à partir d'une source XML et de son fichier de schéma XSD associé. La résultante de cette opération est la création d'instances d'objets utilisant des types primitifs (cf. Annexe 8), ce qui n'est pas satisfaisant si on veut une architecture fortement typée métier. La deuxième étape est la bascule des informations de ces objets créés dans notre structure d'objets fortement typés (cf. Annexe 11).

6.2.3.1 Détail d'un fichier de description

Reprenons balise par balise la description de l'interface au travers du fichier correspondant à l'interface professionnelle standard. Cette description fait référence à la capture d'écran de QB_Utilisateur Version 5 en Annexe 4.

- ***Balise « general »***

Lien vers l'image utilisée dans QB_Utilisateur pour le choix de l'interface.

```
<photo>standard.png</photo>
```

Jeu de couleurs (thème) utilisé pour afficher les différents composants dans l'application.

```
<theme>Office2010Black</theme>
```

Appartenance de l'interface à une version professionnelle ou à une version éducation.

```
<build>Pro</build>
```

Liste des différents liens de menus affichés en bas à droite de l'application, cf. zone 6 de la capture d'écran. Les liens sont des constantes vers des actions prédéfinies dans QB_Utilisateur. Il existe à ce jour 7 types d'actions possibles. Les chaînes de texte à afficher dans l'outil sont renseignées dans les fichiers de traduction.

```
<menu>
        <menu_item num="1">
                <menu_action>CREER_SESSION</menu_action>
        </menu_item>
        <menu_item num="2">
                <menu_action>RECHERCHER_SESSION</menu_action>
        </menu_item>
        ……
```

```
</menu>
```

Permet d'afficher ou masquer les onglets de l'interface.

```
<onglet>
        <onglet_page num="1">
                <onglet_visible>true</onglet_visible>
        </onglet_page>
        <onglet_page num="2">
                <onglet_visible>false</onglet_visible>
        </onglet_page>
……..
</onglet>
```

- ***Balise « session »***

La balise session contient la liste des différents champs du niveau session à afficher en en-tête dans QB_Utilisateur. (cf. zone 2 de la capture d'écran).

```
<session>
        <champ_session num="1">
                <nom_champ_session>LIB_SESSION</nom_champ_session>
                <controle_session num="1">
                        <obligatoire_champ_session>true</obligatoire_champ_session>
                        <type_controle_session>textbox</type_controle_session>
                        <longueur_controle_session>150</longueur_controle_session>
                        <donnees_controle_session>texte</donnees_controle_session>
                        <propertie_session>Libelle</propertie_session>
                </controle_session>
```

```
        </champ_session>

        ……..

</session>
```

La description d'un item session se fait en deux parties. Dans un premier temps, la balise *nom_champ_session* fait référence au nom du champ de la table *tSession* de la base de données. Dans un second temps on décrit avec les informations du nœud *controle_session* le composant qui accueillera ces données. Dans notre exemple, pour la balise LIB_SESSION qui correspond au nom de la session, on peut comprendre qu'il s'agit d'un champ obligatoire (il sera marqué d'un astérisque rouge sur l'interface), que c'est un champ de type texte, que l'affichage de la donnée se fera dans un contrôle de type zone de texte d'une longueur de 150 pixels et qu'il est lié à la propriété *Libelle* de l'objet métier *SessionActive.* (cf. Annexe 5).

- ***Balise « personne »***

Cette balise décrit les différents champs du niveau personne qui servent à construire la grille de saisie des participants. (cf. zone 3 de la capture d'écran). Dans notre exemple, j'ai isolé la balise qui décrit la colonne « Société ».

```
<personne>

        ……..
```

Il s'agit de la dixième colonne de la grille.

```
        <champ_personne num="10">
```

Spécifie la constante à chercher dans les fichiers de traduction pour l'affichage de l'en-tête de la colonne.

```
            <nom_champ_personne>SOCIETE</nom_champ_personne>
```

Cette information n'est pas obligatoire lors de la saisie d'un participant.

```
            <obligatoire_champ_personne>false</obligatoire_champ_personne>
```

Cette colonne est disponible sur le mode saisie grille mais pas en mode fiche.

```
<mode_grille_fiche_personne>grille</mode_grille_fiche_personne>
```

Description du composant qui affiche le nom de la société.

```
<controle_personne>
```

C'est un contrôle de type *dropdownlist*, nom porté par la liste déroulante en .Net. Plusieurs autres valeurs sont possibles comme *checkbox*, *maskbox*, *textbox* ou bien *dropdown*.

```
<type_controle_personne>dropdown</type_controle_personne>
```

Le contrôle fait 90 pixels de largeur.

```
<longueur_controle_personne>90</longueur_controle_personne>
```

On peut stocker du texte, par opposition à « date » ou « numerique » :

```
<donnees_controle_personne>texte</donnees_controle_personne>
```

Cette colonne est liée à la propriété *NomSociete* de notre objet métier *PersonneSessionActive*.

```
<propertie_personne>NomSociete</propertie_personne>
```

Les données qui alimentent le contrôle proviennent de la base de données (BDD), les autres valeurs sont LISTE et PLAGE, avec dans ces deux cas, des paramètres supplémentaires.

```
<type_datasource_personne>BDD</type_datasource_personne>
```

La valeur qu'on va récupérer après la saisie ou la sélection de la société d'un stagiaire est celle du champ *NumSociete* de la table *tSociete*.

```
<valeur_datasource_personne>tSociete.NumSociete</valeur_datasource_personne>
```

Les valeurs qu'on souhaite afficher dans la liste déroulante proviennent du champ *NomSociete* de la table *tSociete*.

```
                <affichage_datasource_personne>tSociete.NomSociete</affichage_datasource_pers
onne>
                </controle_personne>
        </champ_personne>
        ……..
</personne>
```

6.2.3.2 Schéma de principe

Ci-dessous, un schéma décrivant le fonctionnement du chargement des interfaces.

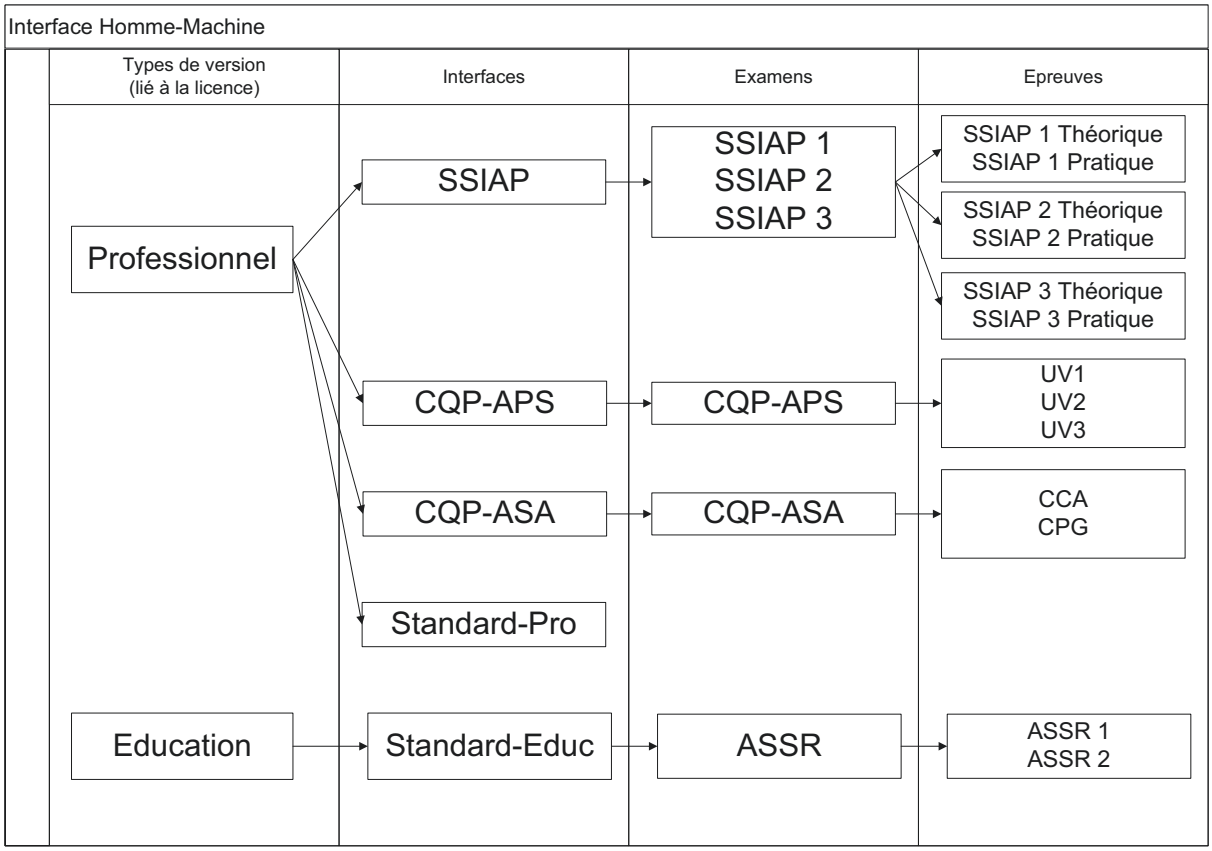


Fig. 9 – Schéma de principe des interfaces

Types de version : A l'installation, le numéro de licence défini le contexte global d'utilisation. Il existe 4 types de licence différentes ; une licence professionnelle et une autre pour le monde de l'éducation et pour ces 2 cas une version illimitée ou limitée dans le

temps. Nous avons développé un outil de génération de numéro de licence en fonction de ces paramètres.

Interfaces : Suivant la version de QuizzBox installée (Professionnelle ou Education), et aussi en fonction du master généré avec InstallShield, on a accès à des interfaces différentes. On peut changer d'interface via la liste déroulante en haut à gauche de QB_Utilisateur. La description de ces interfaces est faite dans des fichiers XML que l'on trouve dans *AppData_Interfaces*. A chacun de ces fichiers XML est rattaché :

- un fichier *interface.xsd*, qui contient la définition de la structure des fichiers XML d'interface ;
- un fichier de traduction *NOM_INTERFACE-lang.xml* dans le répertoire de la langue (fr-fr ou en-us par exemple).

Examens : Un fichier examen décrit les différentes épreuves disponibles pour cet examen. La description d'un examen se fait au travers d'un fichier XML qui liste les épreuves, leur type (pratique, théorique), le questionnaire associé. Il détermine aussi s'il existe un mode entraînement pour ces épreuves, si les épreuves doivent être passées de manière séquentielle.

A chaque fichier d'examen est associé :

- un fichier *examen.xsd*, qui contient la définition de la structure des fichiers XML d'examen.

Epreuves : Une épreuve correspond à un questionnaire à lancer avec QuizzBox ou avec QB_Pratique pour la saisie de notes pour des épreuves de type pratique.

6.3 QuizzBox en version multi-langues

6.3.1 Cas d'utilisation

QuizzBox est aujourd'hui vendu dans plusieurs pays non francophones (Royaume-Uni, Allemagne, Liban…). Il est bien sûr évident que la langue de notre suite logicielle doit

s'adapter en fonction de la langue d'utilisation. Les langues disponibles dans la version précédente étaient le français, l'anglais et l'allemand.

6.3.2 Problématique

En version 4, les chaînes de traduction étaient stockées dans des fichiers RESX, un par langue, qui sont des fichiers XML qui après compilation deviennent des fichiers DLL. Ces fichiers se nomment tous *QuizzDll.resources.dll*, ils sont stockés dans un répertoire nommé du code ISO de la langue cible. On parle de DLL satellites à notre librairie Quizzdll.dll, qui a entre autre la charge via des classes publiques, d'exposer la traduction à nos autres applications. Il s'agit là du processus natif dans l'éditeur Visual Studio .Net pour internationaliser son programme. Nous avions deux problèmes avec ce fonctionnement.

Le premier problème est qu'il était impossible après compilation du programme de modifier les chaînes de traduction, celles-ci étant disponibles au travers des fichiers binaires DLL. Cela enlève de la souplesse sur la personnalisation « à chaud » de l'application. En cas d'erreur sur une traduction, ou si nous devions ajouter une nouvelle langue, il fallait recompiler QuizzDll.dll qui est la librairie.

Le deuxième problème est que la traduction est faite par une société spécialisée qui nous facture la traduction au nombre de mots et que cela représente un coût. Dans le cadre du SSIAP, à titre d'exemple, il est inutile de traduire les chaînes liées à cet examen dans toutes les langues, car c'est un examen franco-français. On ne pouvait pas aisément dans les fichiers RESX faire la distinction entre les chaînes à traduire et les chaînes qui ne devaient pas l'être. Nous n'avons pas la maîtrise de la structure de ces fichiers, il s'agit de fichiers XML avec une balise principale *resources* puis autant de nœuds *string* qu'il y a de chaînes de texte à traduire. Les balises *string* disposent d'un attribut et d'une valeur. Nous faisions la distinction en codifiant l'attribut, alors que la valeur contenait la chaîne de texte traduite. Par exemple, le code QBU_FRMMAIN_LBLRECEPTEUR_CONNECTE renvoie à la chaîne de traduction qui concerne le label (composant texte non modifiable, libellé) se nommant

lblRecepteur du formulaire *frmMain* de l'application QB_Utilisateur pour afficher que le récepteur est connecté.

6.3.3 Réflexion et solution

Suite aux différentes problématiques évoquées ci-dessus, il apparaît qu'il faut d'une part segmenter les chaînes à traduire en fonction du cas d'utilisation et d'autre part que ces fichiers soient éditables post installation.

Concernant la segmentation des chaînes, j'ai choisi de gérer la traduction de QuizzBox sur 3 niveaux. Tous les fichiers de traduction se situent dans le dossier *AppData_Interfaces\CodeLangue* où *AppData* est le répertoire choisi à l'installation pour stocker les données et *CodeLangue* correspond au code ISO de la langue du poste (fr-FR pour le français par exemple). On stocke toutes les chaînes de traduction dans une collection qu'on alimente par la lecture de fichiers XML dans l'ordre suivant :

- *ALL-lang.xml* fichier unique qui contient l'ensemble des chaînes de traduction de l'application.
- *TYPE_VERSION-lang.xml*, où TYPE_VERSION correspond au nom de la version qui a été installée (fonction du numéro de licence). Ces fichiers contiennent toutes les traductions spécifiques à ce type de version (professionnel, éducation). Par exemple, dans le cadre de l'éducation, la chaîne LIB_SESSION sera traduite par classe, alors qu'en version standard professionnelle ce sera session.
- *NOM_INTERFACE-lang.xml*, où NOM_INTERFACE correspond au nom de l'interface en cours d'utilisation. Ces fichiers contiennent les traductions spécifiques aux interfaces. Dans le cadre d'un examen CQP-APS par exemple, on retrouve la notion de numéro d'agrément du centre formateur.

Prenons l'exemple de la chaîne LIB_FORMATEUR. Cette chaîne sera définie dans le fichier de traduction de la version standard professionnelle *standard-lang.xml* avec comme traduction, pour la version française, « Formateur ». Cette même chaîne sera traduite par « Enseignant » dans le fichier *education-lang.xml* stocké dans *AppData\fr-FR*. Dans le cadre de l'habilitation électrique pour Bureau Veritas et de l'interface TimsEvalHE associée, on parle d'« Animateur », cette chaîne sera enregistrée dans le fichier *TimsEvalHE-lang.xml.* Plaçons-nous dans le cas du chargement de l'interface TimsEvalHE ; le système va donc dans un premier temps alimenter la collection des chaînes de traduction avec la valeur de la balise LIB_FORMATEUR du fichier *ALL-lang.xml*. Dans notre cas elle n'existe pas. Puis l'item de la collection va être remplacé par la valeur de cette même balise du fichier *standard-lang.xml* puisqu'on se trouve dans un cadre professionnel (par opposition à éducation) pour finalement être écrasé par la valeur de la balise du fichier TimsEvalHE.

Ces nouveaux fichiers XML de traduction ont la même structure que les fichiers RESX de la version 4. Mais maintenant, ils sont modifiables même après avoir installé QuizzBox. De plus, il est quand même possible de segmenter les différentes chaînes à l'intérieur d'un fichier de traduction en ajoutant des nœuds supplémentaires au fichier. Pour reprendre l'exemple du composant *lblRecepteur*, on pourrait trouver dans le fichier de traduction :

```
<resources>
        <QB_Utilisateur>
                <frmMain>
                    <string="LBL_RECEPTEUR_CONNECTE">Récepteur connecté<string>
                    ……..
                </frmMain>
                ……..
        </QB_Utilisateur>
        ……..
</resources>
```

6.4 Gestion des participants

6.4.1 Cas d'utilisation

L'ancienne version de QuizzBox permet de gérer les participants à la maille session. Une personne qui après avoir passé un examen SSIAP se présente pour passer un examen CQP-APS est vue par le système comme une nouvelle personne. Seule notre application de consolidation des résultats, QB_Consolidation, disponible seulement pour quelques clients, permet de faire le lien entre ces deux personnes. Ce regroupement se fait sur les informations de nom, prénom et date de naissance avec un algorithme de détection des candidats. Si par exemple une madame Dupont vient se présenter une seconde fois dans le centre d'examen et que le formateur se trompe dans l'orthographe de son nom ou si cette madame Dupont a changé de nom depuis (mariage par exemple), le système n'est pas capable de faire le rapprochement entre ces deux personnes. Pour pallier le problème nous avons mis en place une fonctionnalité de gestion des conflits dans QB_Consolidation qui propose, pour des cas où deux personnes ont 2 informations en commun du trio nom, prénom, date de naissance de les regrouper. Cette gestion des conflits est efficace mais contraignante, et nécessite une validation humaine.

Un autre point important soulevé par cette évolution est l'inscription des stagiaires aux stages. En effet, dans le cadre d'un examen HE34-2 par exemple, qui est de type habilitation électrique, les stagiaires sont inscrit à ce stage, mais n'ont pas tous les mêmes acquis ni les mêmes besoins et par conséquent ne passent pas tous les mêmes épreuves. En général sur ce type d'examen, il y a un tronc commun à la formation auquel tous les participants assistent, puis ils doivent passer ensuite les tests théoriques et pratiques où ils sont inscrits.

Cette évolution en version 5 doit donc permettre d'une part de pouvoir gérer l'unicité des participants directement dans QB_Utilisateur et d'autre part d'apporter une réponse concernant les inscriptions des stagiaires.

6.4.2 Problématique

La problématique est de savoir comment QB_Utilisateur va détecter si la personne existe déjà dans le système lors de la saisie d'un participant sur la feuille de présence. Reprenons le cas de madame Dupont. Si le formateur saisit sur la grille de QB_Utilisateur le nom Dupont, le système doit lui proposer la liste des Dupont déjà inscrits sur des sessions précédentes. Dans notre cas, c'est son deuxième passage dans le centre d'examen, le formateur sélectionne la bonne personne dans la liste proposée. Que doit-il se passer si le formateur doit corriger sa date de naissance qui est erronée ? Le système doit-il créer une nouvelle entité personne dans QuizzBox ? Doit-il modifier les informations de madame Dupont ? Doit-il aussi modifier l'historique des résultats des sessions passées ?

Concernant l'inscription des stagiaires, QuizzBox sait gérer jusqu'à présent la non-participation d'un stagiaire à un test théorique en le notant comme absent au test. La vision est encore plus macroscopique puisqu'en spécifiant un stagiaire absent au test qu'on va lancer, on le rend absent à la session et donc à tous les tests suivants. Il y a en effet un lien étroit entre stagiaire et session dans la base de données. Le formateur doit lors du test suivant spécifier de nouveau la présence du stagiaire pour que celui-ci puisse voter. L'évolution doit donc apporter une séparation entre stagiaire et session pour qu'un stagiaire puisse participer à un ou plusieurs tests sur une ou plusieurs sessions.

6.4.3 Réflexion et solution

La seule manière de pouvoir identifier une personne de manière unique dans le système est de lui attribuer un identifiant unique. J'ai choisi d'utiliser un identifiant de type GUID (Global Unique IDentifier) car le langage .Net gère nativement ce type d'identifiant et propose des méthodes pour générer ces codes. Un GUID se présente sous cette forme : {4F3502E0-6F93-11D4-9A0E-0335A62C3401}. La clé primaire de la table *tPersonne* aurait pu servir d'identifiant unique, elle est de type auto-incrément dans la base de données Access, mais l'unicité ne se serait fait que localement sur un poste. Il est nécessaire d'utiliser un identifiant qui soit unique quel que soit l'ordinateur utilisé. En pensant aux évolutions à

venir, c'est ce GUID qui servira à faire le rapprochement des personnes dans les autres programmes de la solution.

Lors de l'ouverture de QB_Utilisateur, ce dernier lit les enregistrements de la table *tPersonne* et créé autant d'instances de l'objet *PersonneSessionActive* dont la propriété *Key* est positionnée avec le GUID généré, qu'il y a de participants sur la session active. La création de ces objets alimente automatiquement la classe *ListePersonneSessionActive* qui facilite la gestion de ces instances par des méthodes natives en .Net de tri, de recherche, etc… Comme évoqué dans le paragraphe sur l'architecture métier, le composant .Net, le *gridview*, qui gère la feuille de présence est liée (databind) à la base de données Quizz.mdb locale. Par conséquent, toute modification, création ou suppression de personnes dans le *gridview*, se propage dans les instances de *PersonneSessionActive* et donc en cascade dans la table *tPersonne*. La liaison automatique entre le gridview et la base de données est obligatoire. L'alternative consistant à charger manuellement les données des personnes dans le *gridview*, et gérer les modifications de ce dernier dans le code entraine d'une part une augmentation du code à produire, et donc des sources d'erreurs. D'autre part, ce fonctionnement exclut la vision métier des personnes, on perd du sens, et surtout du confort dans le développement (absence de complétion automatique par exemple). L'alternative intermédiaire aurait pu consister à lier le *gridview* à la base de données sans passer par les objets métiers, mais pour les mêmes raisons, ce n'est pas une solution viable.

La gestion des inscriptions a été résolue en implémentant au niveau des examens la notion de niveau, notion que je détaillerai dans le chapitre suivant. Dans QB_Utilisateur, l'onglet examen affiche un composant de type *gridview* qui reprend la liste des participants à laquelle s'ajoute une colonne « Niveau d'inscription » ainsi qu'une colonne par épreuve de l'examen. Cette nouvelle colonne propose dans une liste déroulante la liste des niveaux d'inscription définis dans le fichier d'examen. Ainsi le niveau HBC1, pour un examen d'habilitation électrique HE34-4, spécifie que le stagiaire va participer au tronc commun des trois premiers tests, puis seulement aux tests 5 et 6. Le niveau HBC2 comprend lui aussi le tronc commun, et seulement l'épreuve numéro 4. Dorénavant, la gestion de la participation

s'effectuera au niveau de l'examen et plus au niveau du test. De plus, le *gridview* permet visuellement par un code couleur d'avoir une vision synthétique de qui participe à quoi.

Je n'ai pour l'instant pas résolu le problème du rapprochement d'une personne saisie dans la feuille de présence avec cette même personne inscrite à une autre session. Ce point n'est pas inscrit au cahier des charges de Bureau Veritas, en tout cas pour la phase 1, tant que les informations restent en local sur le poste des formateurs. Il s'agit cependant d'une fonctionnalité critique pour laquelle il faudra rapidement apporter une réponse.

6.5 Les examens

La notion d'examen est une des évolutions majeures à intégrer à cette version 5. En effet, jusqu'à présent, le formateur pouvait faire passer différents tests à ses stagiaires et certains de nos modèles de résultats étaient capables de faire le regroupement des résultats sur plusieurs tests mais pas de façon satisfaisante. Chacun des tests passés était indépendant des autres. Il n'était pas possible non plus d'instaurer des règles de validation sur un examen.

6.5.1 Cas d'utilisation

Prenons le cas d'un examen de type habilitation électrique appelé *Inter Initial Elec ;* c'est le type d'examen le plus complexe à mettre en œuvre. Cet examen est composé de 14 tests théoriques et d'un test pratique. Les 4 premiers tests théoriques doivent être passés obligatoirement par tous les stagiaires, il s'agit du tronc commun à la formation. Les suivants, sont passés en fonction de l'inscription des stagiaires à cet examen. Ici, on ne parle pas de réussite à un examen, mais d'habilitations obtenues en fonction des résultats. Ainsi un stagiaire, peut obtenir en fin d'examen les habilitations B0, B1 et B2 alors qu'un autre aura B0, B1V et B2V.

6.5.2 Problématique

Plusieurs problèmes ont été soulevés lors de l'analyse de ce point.

Le premier est qu'un examen, d'un point de vue système, regroupe un certain nombre d'épreuves pratiques ou théoriques soumises à des règles de validation. Il nous faut stocker la définition de ces examens.

De plus, ces examens sont vus comme du contenu associé au logiciel QuizzBox, donc susceptible d'évoluer en fonction de nouvelles règlementations par exemple. Comme la plupart des fichiers de QuizzBox, les examens peuvent être synchronisés depuis un site distant. Imaginons le cas où un formateur lance la première épreuve d'un examen qui en comporte trois, les stagiaires répondent aux sollicitations du système et terminent l'épreuve. Si le formateur, avant de lancer la deuxième épreuve, synchronise les contenus de son QuizzBox avec le référentiel qui se trouve sur un site distant et que les épreuves qui composent l'examen ont changé, le système va se retrouver dans un état instable. En effet, le système, pour présenter les résultats fait un rapprochement entre la description des tests et les résultats associés dans la base de données. Nous devons aussi être capables de pouvoir ajouter, modifier et supprimer des examens/épreuves sans avoir besoin d'installer une nouvelle version de la suite logicielle.

6.5.3 Réflexion et solution

De le même manière que pour les interfaces QuizzBox, et pour les mêmes raisons, le choix de la technologie pour stocker ces descriptions s'est rapidement porté sur le format XML. Ci-dessous, un exemple de fichiers XML d'examen.

Référence au fichier de schéma *examen.xsd.*

```
<examen xmlns="http://tempuri.org/examen.xsd">
```

Version du fichier d'examen.

```
<version>1.0</version>
```

Lien vers le fichier qui décrit l'interface d'affichage.

```
<interface>TimsEval-CACES.xml</interface>
```

Définit si on offre la possibilité d'avoir un mode de type entraînement.

```
<mode_examen>false</mode_examen>
```

Définit si les épreuves doivent être passées séquentiellement, dans un ordre défini.

```
<epreuves_sequentielles>false</epreuves_sequentielles>
```

Définit le modèle de feuille de présence à afficher par défaut pour cet examen.

```
<modele_presence></modele_presence>
```

Définit le modèle de résultats collectifs à afficher par défaut pour cet examen.

```
<modele_collectif>Attestation_Col_CACES_R386.xls</modele_collectif>
```

Définit le modèle de résultats individuels à afficher par défaut pour cet examen.

```
<modele_individuel>Attestation_Indiv_CACES_R386.xls</modele_individuel>
```

Les balises suivantes définissent des niveaux d'inscription. Un stagiaire inscrit à un Niveau1 par exemple, ne doit passer que les épreuves 1 et 3, alors qu'un autre inscrit au Niveau2 doit assister à l'ensemble des tests de l'examen. Les noeud1, noeud2 et noeud3 ne sont présents que pour ajouter de l'information qui sera affichée sur les différents états proposés par le système.

```
<niveaux>
        <niveau nom="Niveau1">
                <noeud1 nom="duree_theorie">2.5 jours</noeud1>
                <noeud2 nom="duree_pratique">0.5 jour</noeud2>
                <noeud3 nom="duree_totale"></noeud3>
        </niveau>
        <niveau nom="Niveau2">
                <noeud1 nom="duree_theorie">3 jours</noeud1>
                <noeud2 nom="duree_pratique">1 jour</noeud2>
                <noeud3 nom="duree_totale"></noeud3>
```

```
        </niveau>
    </niveaux>
```

La règlementation sur le CACES impose aux centres de formation une limite maximum du nombre de participants aux tests en fonction de leur catégorie. On distingue deux catégories, une première qui sera identifiée x par la suite et qui comporte le test *Pratique R386 1A* et une deuxième identifiée y composée des tests *Pratique R386 1B* et *Pratique R386 3A*. La balise suivante, *correspondance*, impose la limite de 6 tests maximum de deuxième catégorie si on ne fait pas de test de la première, 5 tests si on en fait 1, etc…

```
    <limitation>
        <categorie num="1">Pratique R386 1A</categorie>
        <categorie num="2">Pratique R386 1B|Pratique R386 3A</categorie>
        <correspondance num="1" x="0" y="6" />
        <correspondance num="2" x="1" y="5" />
        <correspondance num="3" x="2" y="4" />
        <correspondance num="4" x="3" y="3" />
        <correspondance num="5" x="4" y="2" />
        <correspondance num="6" x="5" y="1" />
        <correspondance num="7" x="6" y="0" />
    </limitation>
```

Définit quels seront les boutons à afficher en zone 5 (cf. Annexe 4, QB_Utilisateur v5).

```
    <epreuve>
```

Cette balise ne porte pas l'attribut *num*, il ne s'agit donc pas d'une épreuve.

```
        <epreuve_bouton>
```

Texte affiché sur le bouton

```
            <nom_epreuve>Fiche de vérification</nom_epreuve>
```

Si nous disposions de plusieurs balises *questionnaire_epreuve* pour le nœud *epreuve_bouton*, il y aurait un tirage aléatoire lors de la sélection du contenu.

```
<questionnaire_epreuve num="1">
```

Chemin relatif depuis un répertoire QuizzBox vers le fichier à lancer.

```
<chemin>\CACES\Fiche_verification _CACES.doc</chemin>
</questionnaire_epreuve>
```

Cette épreuve est rattachée au Niveau1 et au Niveau2

```
<niveau_epreuve num="1">Niveau1</niveau_epreuve>
<niveau_epreuve num="2">Niveau2</niveau_epreuve>
</epreuve_bouton>
```

Cette balise possède un attribut *num* dont la valeur est 1. Cela signifie qu'il s'agit de la première épreuve de l'examen.

```
<epreuve_bouton num="1">
<nom_epreuve>Théorie R386</nom_epreuve>
```

Définit si l'épreuve est obligatoire.

```
<obligatoire_epreuve>false</obligatoire_epreuve>
```

Définit le type du test (théorique ou pratique).

```
<type_epreuve>theorique</type_epreuve>
```

Nous avons ici 3 nœuds *questionnaire_epreuve*. Lors du lancement du test théorique, le système fera un tirage aléatoire de l'épreuve à lancer parmi les 3 fichiers listés ci-dessous.

```
<questionnaire_epreuve num="1">
```

```
                <chemin>CACES\R386\Theorie_R386_jeu_1.ppt</chemin>
            </questionnaire_epreuve>
            <questionnaire_epreuve num="2">
                <chemin>CACES\R386\Theorie R386 jeu 2.ppt</chemin>
            </questionnaire_epreuve>
            <questionnaire_epreuve num="3">
                <chemin> CACES\R386\Theorie R386 jeu 3.ppt</chemin>
            </questionnaire_epreuve>
```

Ce test fait seulement parti du niveau d'inscription Niveau2.

```
            <niveau_epreuve num="1">Niveau2</niveau_epreuve>
        </epreuve_bouton>
```

Le nœud ci-dessous décrit de la même manière une épreuve de type pratique, où le lien vers le test pointe vers un fichier de type XML.

```
        <epreuve_bouton num="2">
            <nom_epreuve>Pratique R386 1A</nom_epreuve>
            <obligatoire_epreuve>false</obligatoire_epreuve>
            <type_epreuve>pratique</type_epreuve>
            <questionnaire_epreuve num="1">
                <chemin> CACES\Pratique_R386 _A.xml</chemin>
            </questionnaire_epreuve>
            <niveau_epreuve num="1"></niveau_epreuve>
        </epreuve_bouton>
    </epreuve>
</examen>
```

Pour pallier le problème d'une éventuelle mise à jour de la description d'un examen, à sa création, le fichier XML de description de l'examen est stocké dans la base de données de QuizzBox et c'est cette description qui servira tant que l'examen en cours ne sera pas terminé. Même si une synchronisation du système modifie le contenu de l'examen courant, tant qu'il ne sera pas terminé, c'est la description d'origine qui sera utilisée. D'un point de vue technique, le contenu du fichier XML est stocké dans un champ de type mémo, et la fonction de chargement de l'examen prend en paramètre soit un fichier XML soit un flux en provenance du champ de la base de données.

6.6 Les épreuves pratiques

6.6.1 Cas d'utilisation

Dans le cadre de l'habilitation électrique, du CACES, du SSIAP et des différents examens CQP, la loi impose aux candidats de passer des épreuves pratiques. Ces épreuves font partie intégrante de l'examen qui comporte aussi des épreuves de type théorique. Concrètement, le formateur, appelé testeur suivant le contexte, part en extérieur et évalue le candidat en situation sur des critères préétablis. Actuellement, l'évaluation se fait sur papier, et le formateur ressaisit ensuite l'ensemble de ces évaluations dans des fichiers Excel. Ces fichiers Excel donnent ensuite lieu à l'édition de diplôme.

La fonctionnalité à implémenter devra permettre de saisir via QuizzBox les évaluations des stagiaires à ce type de tests pratiques. Les résultats devront être disponibles dans QB_Utilisateur au côté des résultats à d'éventuels tests théoriques. Nous devrons par la suite être en mesure de proposer des états pour impression de ces résultats mais aussi agréger ces informations sous la forme d'examen. L'examen pourra lui aussi donner lieu à l'impression des résultats ou à l'édition d'un diplôme suivant un de nos modèles.

6.6.2 Problématique

Pour le cas Bureau Veritas, les testeurs ont à leur disposition des Tablet PC sous Windows XP. Il nous est demandé que la saisie des résultats de tests pratiques se fasse

directement sur ces Tablet PC. Aucun problème pour nous car QuizzBox est une solution Windows qui peut donc s'installer sur ce type de matériel. Cependant, en mode tablette, un stylet est nécessaire. Il faut faire en sorte que le formateur ait le moins de saisie de texte possible à effectuer (via un clavier virtuel qui apparaît contextuellement), et qu'il ait à sa disposition plutôt des listes de choix, des boutons à cliquer, des cases à cocher. En regardant au-delà de ce client, on peut imaginer que d'autres pourraient vouloir évaluer les stagiaires en se servant d'une tablette de type Android ou iPad, ou même un smartphone.

6.6.3 Réflexion et solution

J'avais pensé ce module d'évaluation de tests pratiques comme une fonctionnalité de QB_Utilisateur. Je me suis rapidement rendu compte qu'il faudrait développer un programme séparé. En effet, suivant le périphérique visé pour la saisie, il faudra redévelopper entièrement le module, en Java pour Android, en Objective C pour les produits Apple. QB_Pratique sera donc un nouvel exécutable de la suite QuizzBox. De plus, si le module n'est pas inclus en standard dans la suite logicielle, il sera possible de vendre cette fonctionnalité comme un module complémentaire à QuizzBox.

Bien que QB_Pratique soit un logiciel séparé de QB_Utilisateur, il faut que les deux applications communiquent. La saisie des participants, des informations de la session, le choix de l'examen et du test pratique à lancer se fait depuis QB_Utilisateur. Ensuite, dans QB_Pratique, on évalue les stagiaires, puis ces évaluations doivent ensuite remonter dans QB_Utilisateur pour qu'on puisse poursuivre l'examen sur d'autres épreuves, éditer les résultats, etc… J'ai choisi de faire communiquer ces deux programmes via des fichiers XML. Outre l'intérêt de la sérialisation de ces fichiers que j'ai pu évoquer déjà plusieurs fois au cours de ce mémoire, le format XML présente dans ce cas précis un autre intérêt, la portabilité. Pour peu qu'il soit encodé en UTF-8, ce type de fichier peut être lu par toutes les plateformes cibles de notre outil (Windows, Android, iOS, …), quelle que soit la langue utilisée.

Dans QB_Utilisateur, après avoir saisi les informations relatives à la session, aux participants, à l'examen, un clic sur un bouton d'épreuve pratique génère un fichier de test pratique. Ce fichier est une copie du fichier XML de description de l'épreuve auquel on a ajouté des nœuds concernant les participants. Parmi ces informations, on retrouve les noms et prénoms des stagiaires, mais aussi le niveau d'inscription pour les tests d'habilitation électrique et les modules sur lesquels ils seront évalués. Il est en effet possible que des stagiaires soient exempts d'une partie du test. Ce nouveau fichier sera stocké dans le dossier *Test_pratique_generes*. Suivant le contexte d'utilisation il sera soit passé en argument au lancement de QB_Pratique soit transféré (par mail, réseau ou autre) au périphérique du testeur de l'épreuve et servira de la même manière de point d'entrée à QB_Pratique.

Le chargement du fichier XML dans QB_Pratique obéit aux mêmes règles que celles établies jusqu'à présent dans les autres fonctionnalités de notre version 5 ; à savoir dé-sérialisation du fichier dans des objets créés automatiquement par VB.Net (cf. Annexe 10) puis chargement de ces objets dans des objets fortement typés (cf. Annexe 6). La saisie des évaluations modifie ce fichier XML en ajoutant un nœud résultat et les évaluations associées à chaque participant. A la fermeture de QB_Pratique, si celui-ci a été lancé sur le même poste que QB_Utilisateur, la description XML de l'épreuve plus les informations de résultats sont chargés dans la base de données de QuizzBox. Ainsi les résultats sont directement accessibles depuis l'interface utilisateur. Si QB_Pratique fonctionnait sur une plateforme déportée, le testeur devrait faire parvenir le fichier XML au responsable de la formation qui se chargera d'importer le fichier dans QB_Utilisateur.

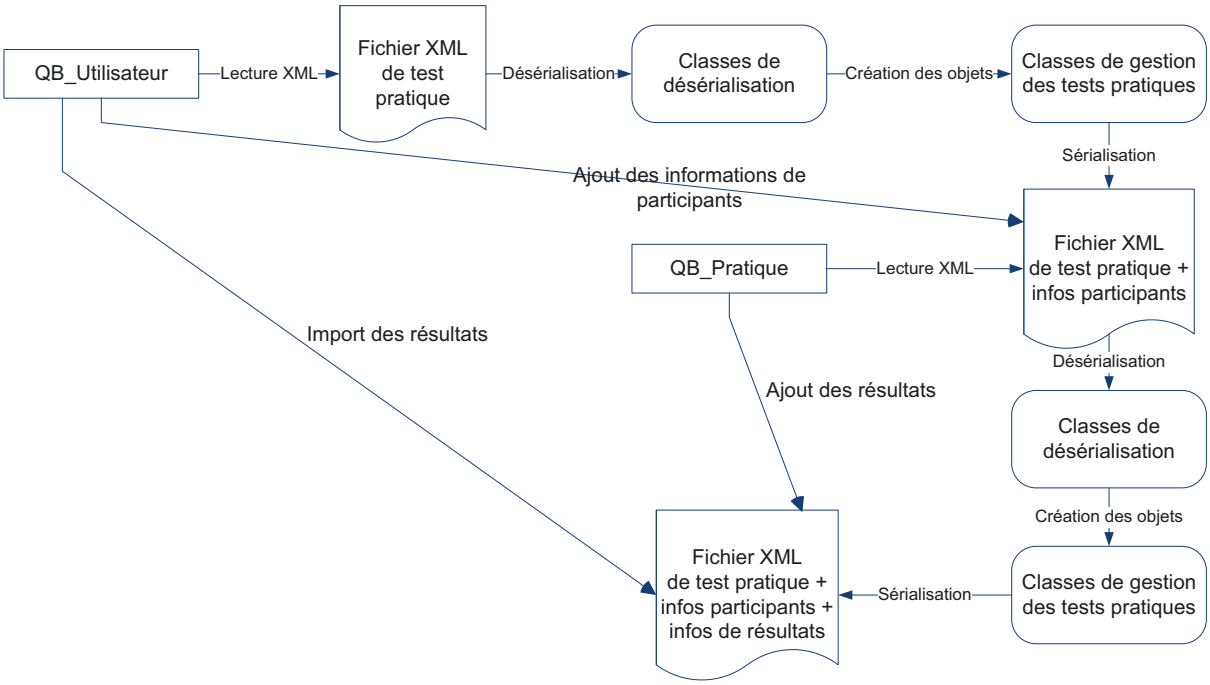


Fig. 10 – Ecosystème de QB_Pratique

6.7 Le récepteur devient multi-applications

6.7.1 Cas d'utilisation

Le récepteur doit être actif à tout moment dès que l'ordinateur de l'utilisateur a démarré. Prenons l'exemple de la télécommande animateur. Si aucun module de QuizzBox n'est lancé, la touche 1 doit permettre de lancer QB_Utilisateur, la touche 2 QB_Administrateur, la touche 3 QB_Questionnaire, etc… Quand QB_Utilisateur est en cours d'utilisation, la touche 1 permet de naviguer dans la liste des questionnaires. Quand le questionnaire est lancé, la touche 1 navigue vers la diapositive précédente. La télécommande administrateur ne doit plus être limitée à l'animation lorsque le test est lancé, mais elle doit être maintenant au cœur de la solution.

6.7.2 Problématique

Le récepteur de la solution QuizzBox est un récepteur radiofréquence qu'on branche sur le port USB d'un ordinateur. Bien que la connectique soit de l'USB, en interne il s'agit d'une interface série.

Les ports série COM dans Windows ne peuvent être ouverts et utilisés que par une seule application à un instant donné. En version 4, le port COM n'était utilisé que par QB_Utilisateur, pour afficher l'état du récepteur, connecté ou non, et par Powerpoint via notre librairie QuizzLib.dll qui a la charge de la réception et du traitement des réponses des candidats. Ainsi, lors du lancement d'un test depuis QB_Utilisateur, ce dernier se fermait avant de lancer la présentation interactive puis se rouvrait à la fin du test.

6.7.3 Réflexion et solution

Il faut que le port COM utilisé par le récepteur soit accessible facilement par tous les programmes de QuizzBox et de manière non exclusive à une seule application comme c'est le cas dans les anciennes versions. La solution retenue est la création d'un service Windows lancé au démarrage du système d'exploitation qui sera chargé de gérer une instance du port série utilisé par le récepteur et de la faire transiter d'une application à une autre suivant le contexte.

Cette solution est d'un point de vue architecture une application client-serveur. Le service Windows fait office de serveur et les autres applications QuizzBox font office de clients. En .Net, la technologie qui permet de faire des applications distribuées se nomme .Net Remoting. Nous aurions pu utiliser un web service, mais ce dernier nécessite l'utilisation d'un serveur type IIS pour fonctionner. .Net Remoting laisse le choix de la communication entre ces différentes applications suivant plusieurs méthodes. Nous avons choisi de faire transiter les données en utilisant les canaux IPC.

Les alternatives aux canaux IPC pour le transit des données sont d'une part HTTP en utilisant le formateur SOAP. C'est la méthode à utiliser dès lors que la communication se fait

par internet. D'autre part, on aurait pu utiliser les canaux TCP avec un formatage binaire comme pour les canaux IPC.

Le choix du protocole IPC a été fait car il est natif dans le framework.Net et est plébiscité par Microsoft pour les développements. De plus les canaux IPC contrairement aux autres technologies citées ne sollicitent pas le pare-feu lors de son implémentation dès lors qu'il s'agit d'une communication entre processus d'une même machine. L'accès au pare-feu pose la problématique des droits sur les postes des utilisateurs.

7 Mise en œuvre et déploiement

7.1 Cahier des charges

La rédaction du cahier des charges a débuté à la fin de l'année 2010, à la demande de Bureau Veritas. J'ai collaboré à la rédaction de ce document écrit en grande partie par Olivier Dusserre, le chef de projet QuizzBox. Je ne savais pas encore que le mémoire que je prépare aujourd'hui porterait sur ce sujet et donc mon implication a été moindre que celle que j'aurais pu espérer. Néanmoins, j'ai quand même apporté au chef de projet des idées, des avis et quelques schémas qui ont été repris pour la formalisation des besoins du client. J'ai aussi rédigé certains paragraphes plus axés sur la technique.

7.2 Analyse

La phase d'analyse de ce projet a débuté en même temps que la rédaction de ce mémoire. Elle a duré plus de 2 mois et il en a résulté :

- une maquette de l'interface de QB_Utilisateur très proche du résultat final. Je l'ai construite de manière itérative avec Olivier et Jean-Marc Nermond, qui ont tous les deux un sens aigu de l'ergonomie. Si tout au long du projet j'ai eu beaucoup d'autonomie et de liberté sur les choix à faire, toute l'équipe a participé à l'élaboration de cette maquette ;
- des spécifications techniques générales. J'ai scindé l'application en plusieurs grosses fonctionnalités, comme celles décrites dans ce mémoire. Pour une majorité de ces fonctionnalités, les spécifications détaillées ont été écrites en faisant du reverse-engineering de la version 4;
- un diagramme de classes conséquent qui regroupe aujourd'hui près de 80 classes. L'établissement de ce document a été sans nul doute la partie la plus intéressante du projet. Il était nécessaire que nous soyons tous les deux, Olivier et moi impliqués dans cette tâche ; de ce diagramme de classes allait dépendre des

années de développement à venir. J'ai travaillé seul sur cette tâche, mais chacune des classes créées, chaque lien, a fait l'objet de débat, de discussion. De cette confrontation d'avis a émergé, je pense, une architecture solide pour notre application ;

- un schéma de base de données mis à jour. L'annexe 12 propose un schéma de la base de données locale Microsoft Access mettant en évidence les différences entre l'ancienne version de QuizzBox et la nouvelle.

La difficulté majeure lors de la phase d'analyse a été de définir le périmètre des évolutions. Dans le cas de la problématique sur la personnalisation de l'interface, fallait-il par exemple mettre en place un outil disponible pour les utilisateurs finaux permettant de créer ou modifier les interfaces ? Non, probablement pas.

7.3 Développement

La partie développement s'est répartie entre Olivier et moi de manière naturelle. Nous avons en effet, depuis les débuts de QuizzBox chacun la charge de fonctionnalités ou modules du projet bien précis.

Olivier gère la partie logicielle de tout ce qui a trait au matériel (récepteur, télécommandes) et à la génération, l'import-export des résultats. Sur ce projet, il a conçu le service qui gère le récepteur et modifié en profondeur le fonctionnement d'édition de résultats pour que cela ne soit plus géré par QB_Utilisateur mais par des librairies tierces dédiées exclusivement à cet emploi.

Pour ma part, j'ai développé QB_Utilisateur et QB_Pratique, puis adapté les autres outils du package destinés aux formateurs.

7.4 Tests

Les tests ont bien sûr été présents tout au long du cycle de création de l'application, que ce soit les tests unitaires ou les tests d'intégration. Le responsable QuizzBox de Bureau Veritas a lui aussi participé aux tests. Il a eu la charge de valider fonctionnellement que notre

produit répondait à tous ses besoins. Par la suite, quand QuizzBox a intégré l'application QB_Pratique, ce sont 6 formateurs de Bureau Veritas qui ont installé cette nouvelle version pour aussi participer aux tests utilisateurs, et le cas échéant nous remonter des observations sur nos outils.

La communication entre Next Media et Bureau Veritas sur les anomalies se fait via le bug-tracker Mantis qui est un portail Internet permettant la saisie de bugs, l'assignation de tâches à un développeur, etc… Cette solution a été mise en place par le service informatique de Bureau Veritas ; c'est ainsi qu'ils fonctionnent avec leurs différents prestataires. Cette plateforme est aussi utilisée pour les demandes d'évolution.

L'expérience de mes emplois passés et des développements effectués au sein de Next Media m'ont appris à ne pas sous-estimer les différentes étapes de tests. Même si sur ce projet je n'ai pas utilisé de logiciels dédiés aux tests d'applications, j'ai néanmoins été rigoureux sur la méthodologie de ces derniers.

J'ai mis en place lors du développement la gestion d'un fichier journal de type log pour faciliter les étapes de test. Chaque fonction, procédure critique du code est bornée par des instructions *try/catch*, connues aussi des développeurs Java. Le code du bloc *try* est exécuté entièrement sauf si une erreur survient. Dans ce cas, le programme lève une exception et continue dans le ou les blocs *catch* associés. C'est dans cette partie du code qu'on essaie de résoudre le problème et/ou qu'on affiche un message à l'utilisateur et/ou qu'on écrit dans le fichier journal. Pour chaque problème dans le fichier journal, on retrouve la date et l'heure à laquelle l'erreur est survenue, le nom de la fonction et l'endroit où elle se situe dans le code, le type d'erreur, la pile des appels qui ont été effectués avant d'entrer dans cette fonction et même le numéro de la ligne du fichier source où le programme a levé l'exception. Du fait de l'utilisation des instructions *try/catch*, ce journal est vraiment indispensable. Nombre d'erreurs peuvent survenir sans que le testeur ne soit averti visuellement. La gestion des erreurs par un fichier journal permet aussi l'analyse a postériori d'un problème survenu dans l'application.

Il est important aussi lors du développement de passer en pas-à-pas le code qu'on vient d'écrire. Cela permet d'une part d'être sûr que le programme à un moment ou un autre exécute toutes les lignes de code écrites, qu'il n'y ait pas de code orphelin voire du code inutile. D'autre part, cela permet d'observer le comportement du programme par l'affichage par exemple des valeurs des variables à tout moment. Le débogage m'a aussi permis grâce aux points d'arrêt de modifier le contenu de certaines variables afin de me retrouver dans des conditions difficiles à simuler en exécution classique.

Concernant les phases de tests que sont les tests unitaires et d'intégration, il est de mon point de vue difficile d'utiliser des scripts de tests automatiques ou des logiciels dédiés. En effet, il n'y a pas de scénario réel à tester avec un fichier en entrée du programme, un traitement puis un fichier en sortie de l'application. QB_Utilisateur est une application mono-utilisateur dite évènementielle, il nous faut plutôt tester des comportements liés à des actions de l'utilisateur. Le plan de test, non formalisé, consiste à tenter de reproduire toutes les actions possibles de l'utilisateur dans l'interface et de vérifier la remontée des résultats des différents tests dans le système.

7.5 Déploiement

Le déploiement s'est déroulé en plusieurs phases conformément aux documents livrés en début de projet.

- Décembre 2011 : Livraison d'une première version de QuizzBox sans le module QB_Pratique. Cette version est à destination seulement du responsable projet de Bureau Veritas. Elle lui permet de voir quelles sont les voies que nous avons choisies pour cette nouvelle version.
- Janvier 2012 : Livraison de QuizzBox avec le module QB_Pratique toujours pour le responsable projet de Bureau Veritas. L'objectif de cette livraison est de valider avec lui les différents examens d'habilitation électrique et CACES que nous avons préparés.

- Février 2012 : C'est la première version (5.2.0.1) de QuizzBox qui est déployée sur les postes des formateurs du panel de test, au nombre de 6.
- Mars 2012 : QuizzBox intègre désormais les examens de type ASSR. La nouvelle version n'étant pas encore assez éprouvée, nous avons dû faire cette évolution sur l'ancienne version de notre système.
- Avril 2012 : Le panel de test reçoit la version 5.2.0.2. Cette version intègre désormais tous les examens et tests d'habilitation électrique.
- Avril 2012 : Evolution majeure de l'examen CQP-APS. L'organisme en charge de ces examens, l'ADEF, nous a transmis un cahier des charges afin de faire évoluer notre système pour la prise en compte des nouvelles évolutions. Ces examens sont particuliers. En effet les tests sont construits en faisant du tirage aléatoire de questions dans des fichiers Excel chiffrés. Contrairement au chiffrement standard implémenté dans QuizzBox (AES Rijndael 128bits), ces tests sont chiffrés par le biais d'une clé USB de type HASP-HL Nous avions la possibilité de continuer de faire évoluer la version 4 de QuizzBox ou faire en sorte que la version 5 intègre ces évolutions. C'est la deuxième option qui a été choisie. Le système même s'il ne propose pas aujourd'hui toutes les fonctionnalités de la version précédente, contient toutes les fonctionnalités des anciennes versions CQP-APS. Nous disposons maintenant d'une version SSIAP / CQP-APS. Cette nouvelle version concerne plusieurs centaines de nos clients.
- Avril 2012 : Nouvelle version de la solution (5.2.0.3) qui intègre la gestion des signatures électroniques ainsi qu'un module qui existait en version 4. Il s'agit de QB_SaveRestore, un programme qui permet la sauvegarde et la restauration des résultats d'un poste QuizzBox. Cette application est lancée automatiquement à la désinstallation et à l'installation du programme, ce qui permet de retrouver ces données entre deux mises à jour par exemple. Les

testeurs en attendant une version finale pourront dorénavant retrouver leurs données entre deux mises à jour de QuizzBox.

A ce jour, le service Windows qui gère le récepteur n'a pas encore été déployé. En effet, cette évolution n'a pas été demandé par Bureau Veritas ni par un autre client. Il s'agit juste d'un confort à l'utilisation de QuizzBox. Cette fonctionnalité demandera d'être testée sur un panel large de configurations différentes. La mise en place d'un service au sein de Windows peut générer beaucoup de problèmes en fonction des droits de l'utilisateur sur sa machine. Nous savons d'ores et déjà qu'il ne pourra pas être déployé chez certains clients dont le service informatique verrouille au maximum les actions possibles des utilisateurs.

7.6 Relations avec le client

Les relations avec les clients sur les projets sur lesquels j'ai travaillé jusqu'à présent se limitaient souvent au support et au déploiement des applications. Sur la refonte de QuizzBox, j'ai vraiment été un interlocuteur privilégié côté équipe projet. Il n'y a quasiment pas eu un jour sans que j'ai un contact avec Bureau Veritas. Que ce soit par e-mail, par téléphone, via le portail Mantis ou encore en réunion, la communication avec les différents acteurs du projet était indispensable d'un côté comme de l'autre. En effet, en plus des validations périodiques lors du développement, Bureau Veritas a construit ses différents contenus en fonction de nos indications. C'est ainsi qu'aujourd'hui, ils ont à leur disposition dans QuizzBox 15 examens sur le domaine de l'habilitation électrique et 2 sur le CACES qu'ils ont eux-mêmes créés. Ces fichiers XML ont été soumis à validation chez nous, sur la forme, pas sur le fond, avant d'être intégrés dans leur master d'installation.

8 Evolutions futures

8.1 Le projet Bureau Veritas

8.1.1 Point sur la phase 1

La phase 1 du projet visant à la refonte de QB_Utilisateur et l'adaptation des autres programmes de QuizzBox est maintenant sur le point de s'achever. Elle ne se terminera qu'après avoir redéveloppé d'une part QB_SaveRestore, notre programme de sauvegarde et de restauration des données entre deux installations de QuizzBox. D'autre part, il nous faudra intégrer une nouvelle fonctionnalité à QB_Utilisateur qui est la gestion de la feuille de présence avec signature électronique. La feuille de présence en version 4 était un fichier Excel avec la liste des participants que le formateur pouvait imprimer pour faire signer les stagiaires. Combinée avec un ordinateur portable de type Table-PC, cette évolution doit permettre aux stagiaires de signer électroniquement avec un stylet.

8.1.2 La phase 2

La deuxième phase du projet consiste à modifier les autres outils de la solution dont Bureau Veritas souhaite disposer ; les outils de synchronisation et le portail internet QB_Portail.

SynchroQuizz_Admin permettra aux responsables de formation de mettre à disposition de nouveaux contenus et examens ou mettre à jour les fichiers existants sur un serveur FTP. SynchroQuizz_User sera utilisé par les formateurs pour télécharger ces fichiers mais aussi pour envoyer les résultats des sessions passées.

QB_Portail ne sera pas simplement adapté, les besoins exprimés par Bureau Veritas nécessitent de revoir en grande partie cette application. En plus des fonctionnalités actuelles, il souhaite que ses clients puissent accéder au portail et télécharger les feuilles d'émargement et les différentes évaluations de leurs salariés.

8.1.3 Maintenance

Concernant la maintenance du logiciel, tant que les outils de synchronisation et de sauvegarde/restauration ne sont pas à disposition de Bureau Veritas, les mises à jour de QuizzBox devront être téléchargées manuellement par les formateurs. C'est un fonctionnement par patch, un exécutable doit être lancé sur l'ordinateur du formateur pour mettre à jour notre produit. Ce dernier déploie les nouveaux fichiers, remplace ceux que l'on a modifiés, accède éventuellement au registre et met à jour la base de données. Ce système de mise à jour est un outil que nous avons créé dans le cadre d'un autre projet. Par la suite, le responsable projet QuizzBox pourra diffuser ces patches au travers de SynchroQuizz_Admin.

8.2 Les autres projets

A ce jour, nous disposons d'une version de QuizzBox qui peut être déployée chez 95% de nos clients ; les enseignants, les professionnels de la sécurité, les différents types d'animateurs. Pour la plupart, ils n'utilisent que les outils « locaux » de QuizzBox et ne possèdent pas les autres programmes de la solution.

Pour les autres, ceux disposant des outils dits connectés, nous entrerons en mode « projet » avec chacun d'eux pour faire évoluer notre solution. Bien que la solution QuizzBox soit développée pour être réutilisable, chaque déploiement est unique. QuizzBox est pensé pour se trouver au centre du système d'information, quel qu'il soit. Par conséquent, l'intégration au sein de ce système est toujours spécifique.

OPPBTP est un autre de nos clients importants avec près de 100 systèmes QuizzBox. Il dispose déjà depuis de nombreuses années des outils de synchronisation, du portail internet ainsi que de notre outil de consolidation des résultats. Le responsable du projet QuizzBox de cette société a suivi avec attention nos derniers mois de développement mais il n'est pour l'instant pas à l'ordre du jour de migrer la solution mise en place vers la nouvelle version. Nous avons d'ailleurs des demandes d'évolution à faire sur le produit, mais elles seront faites en version 4. En effet, même si nous avions en tête le fonctionnement des outils que l'on a mis en place chez ce client lors du développement de la nouvelle version, la migration

de toute la solution reste néanmoins un processus long et complexe. Il nous faudra repartir sur un mode projet, car bien que nos outils soient réutilisables, certaines fonctionnalités de la solution sont spécifiques à ce client.

Conclusion

La refonte de QuizzBox est un projet critique pour Next Media. En effet, nous avions un produit en version 4 très stable et simple d'utilisation déployé chez plusieurs centaines de nos clients. Cette nouvelle version ne doit pas être un retour en arrière sur la stabilité et doit présenter au moins toutes les fonctionnalités des versions précédentes. Il serait dommageable pour notre entreprise de diffuser massivement un logiciel comportant des « bugs ». Il nous a fallu longtemps pour imposer QuizzBox auprès de nos utilisateurs, il faudrait moins de temps pour ternir notre réputation. Le déploiement chez nos clients va se faire progressivement au cas par cas. Même si aujourd'hui, nous n'avons pas de retours négatifs, bien au contraire, le produit est encore trop jeune pour que nous soyons sûrs qu'il n'y ait pas de régression.

La plupart des objectifs visés ont été atteints. Nous avons aujourd'hui la possibilité de livrer des versions standard, SSIAP-CQP, éducation et Bureau Veritas.

Il y a cependant un objectif secondaire que nous nous étions fixé en début de projet qui n'a pas été atteint. Il s'agissait de limiter le nombre de masters d'installation de notre produit. Le système est aujourd'hui utilisé dans des contextes trop différents. Malgré le nombre de masters différents, le code est lui unique et seule la compilation de certains programmes diffère.

La principale contrainte sur le projet a été le temps. Bureau Veritas nous a commandé, et donc financé des évolutions sur notre produit. La refonte complète du système ne faisait pas partie du cahier des charges du client, c'est donc un coût que nous avons pris en charge. En plus du coût, c'est un temps d'analyse et de développement qu'il a fallu inclure dans le projet en impactant au minimum les délais prévus.

Sur un plan personnel, je suis particulièrement satisfait d'avoir travaillé sur ce projet. Je suis fier de l'architecture créée pour QB_Utilisateur. J'ai déjà dû implémenter des évolutions non prévues au cahier des charges de Bureau Veritas. La facilité de mise en œuvre de ces évolutions me laisse à penser que j'ai fait de bons choix.

De plus, la confiance accordée par Olivier Dusserre et Jean-Marc Nermond a été pour moi l'occasion de montrer ma capacité à prendre des responsabilités qui jusqu'à présent étaient limitées au périmètre de mes développements. C'est avec enthousiasme que j'ai participé et parfois organisé les réunions avec le client, c'était à ce jour un domaine réservé exclusivement au chef de projet.

Aujourd'hui, tous nos espoirs sont tournés vers trois projets. Le premier est le projet ASSR, une version de QuizzBox dédiée aux enseignants afin de faire passer des examens dans le domaine de la sécurité routière à leurs élèves. C'est un marché potentiel de 15000 systèmes de vote interactif. Le processus d'équipement des différents établissements scolaires sera probablement long, mais nous sommes d'ores et déjà prêts à livrer les versions adéquates. Le deuxième projet concerne la société Vinci qui commence à s'équiper progressivement de QuizzBox pour l'accueil sécurité des ouvriers travaillant sur le projet LGV SEA (Ligne Grande Vitesse Sud-Europe Atlantique). Il y a 4500 personnes à former sur ce qui est un des plus importants projets ferroviaires européens actuel. Le dernier projet qui est encore à l'étude implique la société Total. Cela concerne 30 de leurs projets appelés Yards, dénomination correspondant à des projets importants dont les sommes en jeu dépassent le milliard d'euros. Les QuizzBox seraient là aussi utilisés pour l'accueil sécurité du personnel œuvrant sur ces chantiers.

ANNEXE 1 – Le groupe AGINIUS

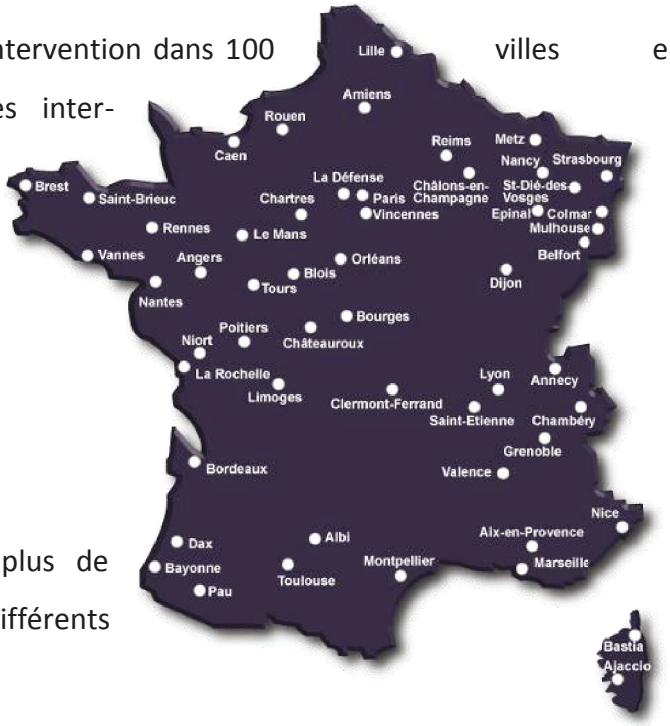
Next Media est cofondateur du réseau national AGINIUS.

Le groupe AGINIUS s'est développé avec les sociétés leader de chaque région dans les domaines de ses expertises informatique, bureautique et multimédia.

Les membres fondateurs ont mis en place toute une palette de services autour de la fonction formation au sein des entreprises, administrations et collectivités locales. Le groupe via ses formateurs peut intervenir partout en France, sur tous les sujets de ses domaines d'expertise, en mettant à disposition son savoir et ses outils, à travers des moyens humains et logistiques importants. Le développement régulier de chacune de ses composantes lui permet de compter aujourd'hui parmi les acteurs majeurs de la formation.

Nos implantations

Proximité avec une capacité d'intervention dans 100 villes en France et 40 centres accueillant les inter-entreprises.



- Capacité logistique

200 salles équipées

- Capacité d'accueil

2000 stagiaires par jour

- Moyens humains avec plus de 300 experts dans différents domaines.

Les formations techniques effectuées auprès des grands éditeurs, le partage de compétences entre nos équipes et les certifications passées par nos formateurs,

garantissent la mise à jour de leur connaissance, l'intégration de nouvelles compétences très pointues dans les domaines des réseaux, du développement et de la bureautique.

- Moyens techniques

Les outils de développement comme Visual Studio ou Eclipse permettant la création d'applications multiplateforme, clients/serveur, n-tiers, web ou riches, l'interconnexion de moyens de communication sont les outils dont disposent les formateurs.

ANNEXE 2 – Agréments et certifications de formation

Pearson VUE est le service d'examens électronique de Pearson Education, le leader mondial de la littérature éducationnelle intégrée. Pearson Education fait partie de Pearson, la société de media internationale en tête du marché de la littérature de formation, d'information professionnelle et des consommateurs.

Pearson VUE implémente des solutions d'examens électroniques améliorant les performances, la fiabilité et la sécurité des programmes de licence et de certification dans le monde entier. Les examens ont lieu sur un réseau privé de plus de 200 centres d'examens professionnels appartenant à et exploités par Pearson et de plus de 3500 centres homologués Pearson VUE dans le monde entier.

Grâce à l'agrément Prometric Test Center, Next Media peut faire passer dans ses locaux les principaux examens des grands programmes de certification informatique du marché, à savoir ceux sanctionnant les différentes certifications Microsoft (Certified Professional, System Engineer, Solution Developer, etc…). Nous avons donc la possibilité de délivrer les certifications individuelles Microsoft.

Les Gold Certified Partners présentent un haut degré de compétences et d'expertise dans les technologies Microsoft et entretiennent des relations de travail étroites avec Microsoft.

Pour ce qui est de Next Media, le domaine d'expertise est la formation. La « Learning Solutions » s'adresse aux partenaires ayant démontré leur expertise dans des offres de formation de haut niveau auprès d'individus et d'organisations, afin d'optimiser les investissements dans les technologies Microsoft. Les partenaires Learning Solutions (Formation) sont à même de conseiller une approche de formation qui s'appuie sur les produits et services Microsoft Learning.

ANNEXE 3 – Le matériel QuizzBox

Le QB1 : QuizzBox 1ère génération

Fig. AN3.1. Télécommande

Fig. AN3.2. Récepteur

Le QB2 : QuizzBox 2ème génération

Fig. AN3.3
Télécommande orange

Fig. AN3.4
Télécommande noire

Fig.AN3.5
Récepteur USB

ANNEXE 4 – Les modules de QuizzBox

La barre d'outils QuizzBox

Ci-dessous une capture d'écran de la barre d'outils QuizzBox, dans Powerpoint 2007.

Sur les versions inférieures (2002, 2003), la barre d'outils se présente différemment, mais possède les mêmes fonctionnalités.

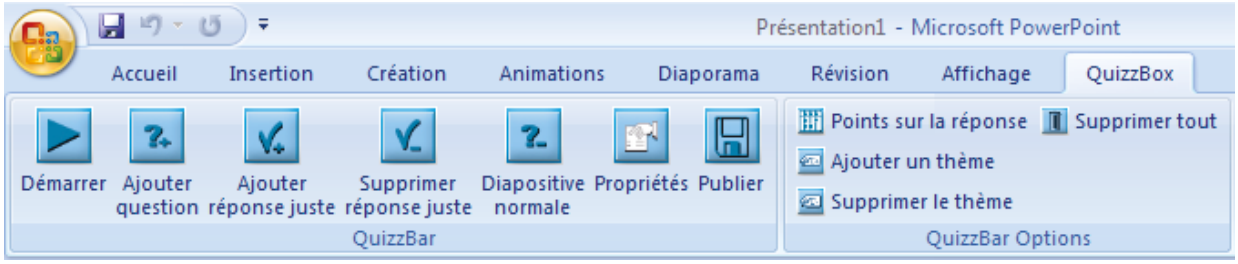


Fig. AN4.1 – Barre d'outils QuizzBox

Lance le questionnaire actif pour le groupe d'utilisateurs actif (le dernier utilisé dans QB_Utilisateur).

Définit la diapositive comme étant une question. Se rajoute automatiquement, si on définit une réponse juste.

Définit une réponse juste. On peut en définir plusieurs. Si aucune ne l'est, il s'agira d'une question de type sondage.

Permet de supprimer une réponse juste.

Définit la diapositive comme une diapositive normale, si auparavant elle avait été définie comme une question ou un thème.

Accède à la liste des propriétés de la diapositive en cours et à celles du questionnaire.

Enregistre le questionnaire, et affiche le rapport d'enregistrement.

Définit un nombre de points pour une des réponses de la question en cours.

Définit la diapositive active comme un thème.

Définit la diapositive active comme normale.

Supprime tous les éléments QuizzBox de la diapositive courante.

QB_Utilisateur Version 4

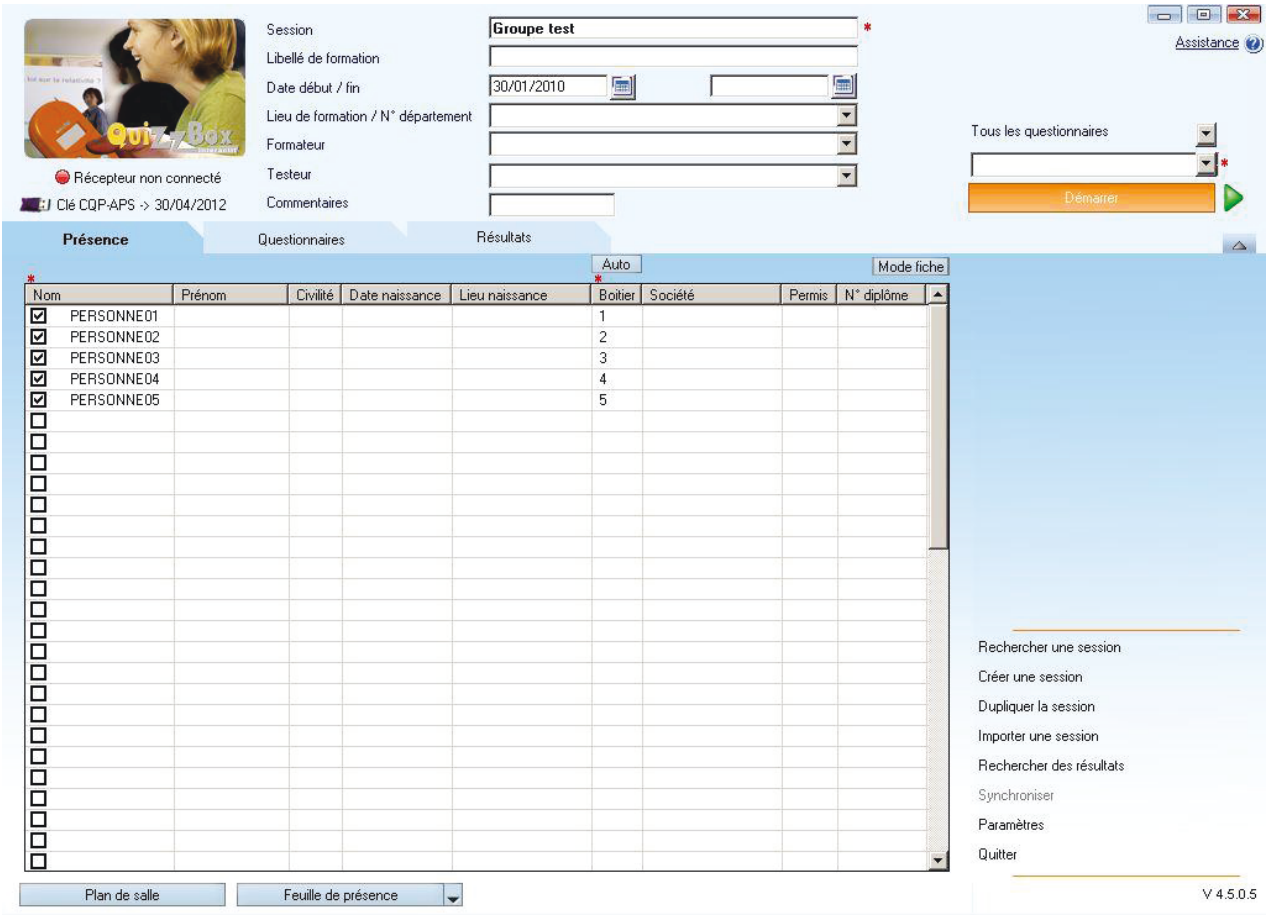


Fig. AN4.2 - Interface QB_Utilisateur V4

QB_Utilisateur Version 5 – Description générale

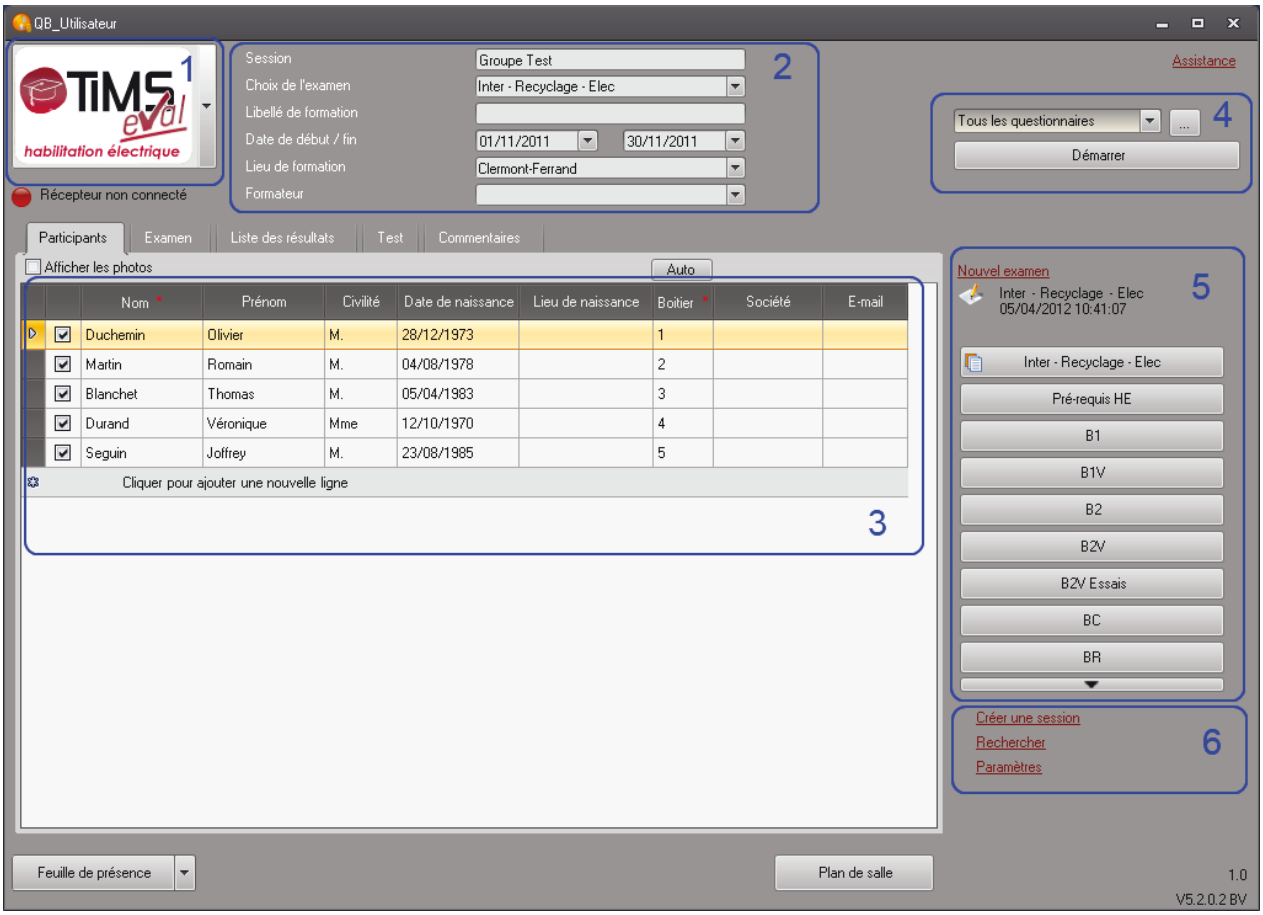


Fig. AN4.3 - Interface QB_Utilisateur V5 – Liste des participants

L'interface est découpée en plusieurs zones :

- Zone 1 : Liste des interfaces. C'est le point d'entrée de l'application qui conditionne le reste de l'affichage. Chaque interface correspond à un cas d'utilisation du système et est décrite dans des fichiers XML.
- Zone 2 : Informations de session. La liste et le type des champs affichés est personnalisable. La description de ces champs se trouve dans les fichiers XML d'interface.
- Zone 3 : Liste des participants. Les participants sont soit saisis manuellement par le formateur soit importés via des fichiers XML de type session. Les

colonnes de la grille de saisie sont personnalisables et leur description se trouve aussi dans les fichiers d'interface.

- Zone 4 : Liste des questionnaires hors examens. Cette liste de questionnaires contient tous les questionnaires utilisables avec QuizzBox qui ne sont pas des épreuves d'examen. On y trouve aussi les questionnaires d'exemple déployés avec l'installation de QuizzBox.

- Zone 5 : Zone de l'examen. On retrouve dans cette zone la date de création et le type d'examen en cours. Les boutons au-dessous permettent de lancer les épreuves de l'examen, ou affichent des contenus de tous types (Pdf, Excel, Word…). La description des boutons d'épreuve se trouve dans des fichiers XML de type examen.

- Zone 6 : Menu de l'application. Le menu de l'application est là aussi fonction de l'interface choisie et par conséquent décrit dans les fichiers XML d'interface.

QB_Utilisateur Version 5 – Onglet examen

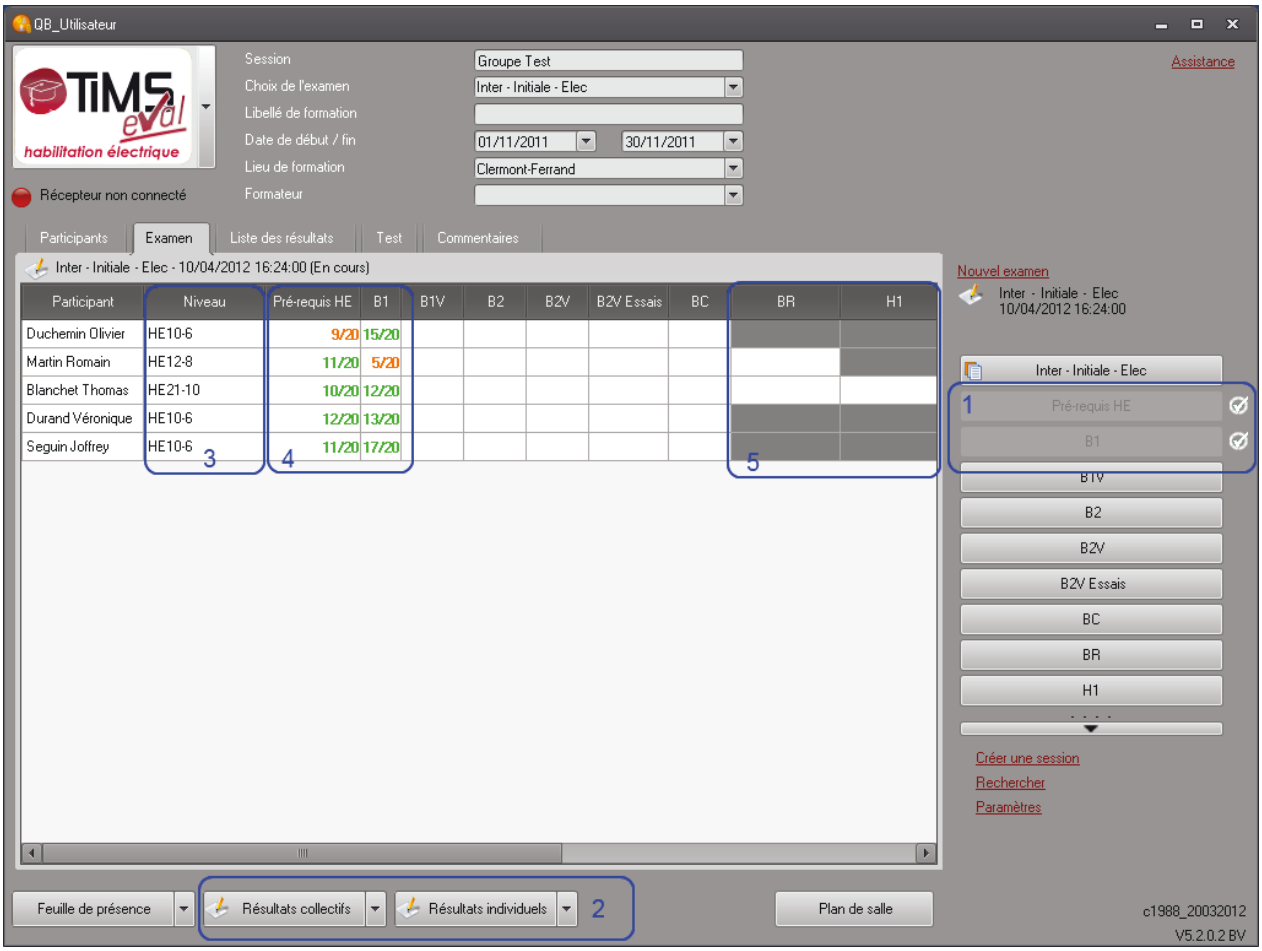


Fig. AN4.3 - Interface QB_Utilisateur V5 – Onglet examen

Par rapport à la capture d'écran précédente, on peut voir d'une part que dans la zone identifiée 1, les boutons d'épreuve « Pré-requis HE » et « B1 » sont grisés et qu'il est affiché un indicateur de progression à droite des boutons signifiant que les tests ont été terminés. D'ailleurs la zone 4 affiche les résultats par candidat à ces deux épreuves. Il est possible d'imprimer ces résultats en cliquant sur les boutons « Résultats collectifs » ou « Résultats individuels » situés dans la zone 2. Ce sont deux boutons munis d'une liste déroulante permettant de choisir le modèle (template) à appliquer aux données de résultat, pour une personnalisation de l'impression.

C'est dans la zone 3 que le formateur inscrit les stagiaires à l'examen. Olivier Duchemin inscrit au niveau HE10-6 ne participera pas aux épreuves BR et H1 visibles dans la

zone 5. Romain Martin est lui inscrit à un niveau différent et ne participera pas à l'épreuve H1. L'affectation des épreuves aux différents niveaux se fait dans les fichiers XML de description des examens.

QB_Administrateur

Fig. AN4.4 - Interface QB_Administrateur

QB_Administrateur est le programme d'administration de QuizzBox. Il permet de modifier les paramètres du système. La plupart de ces paramètres sont stockés dans des fichiers XML et QB_Administrateur est l'interface de modification de ces fichiers.

QB_Questionnaire

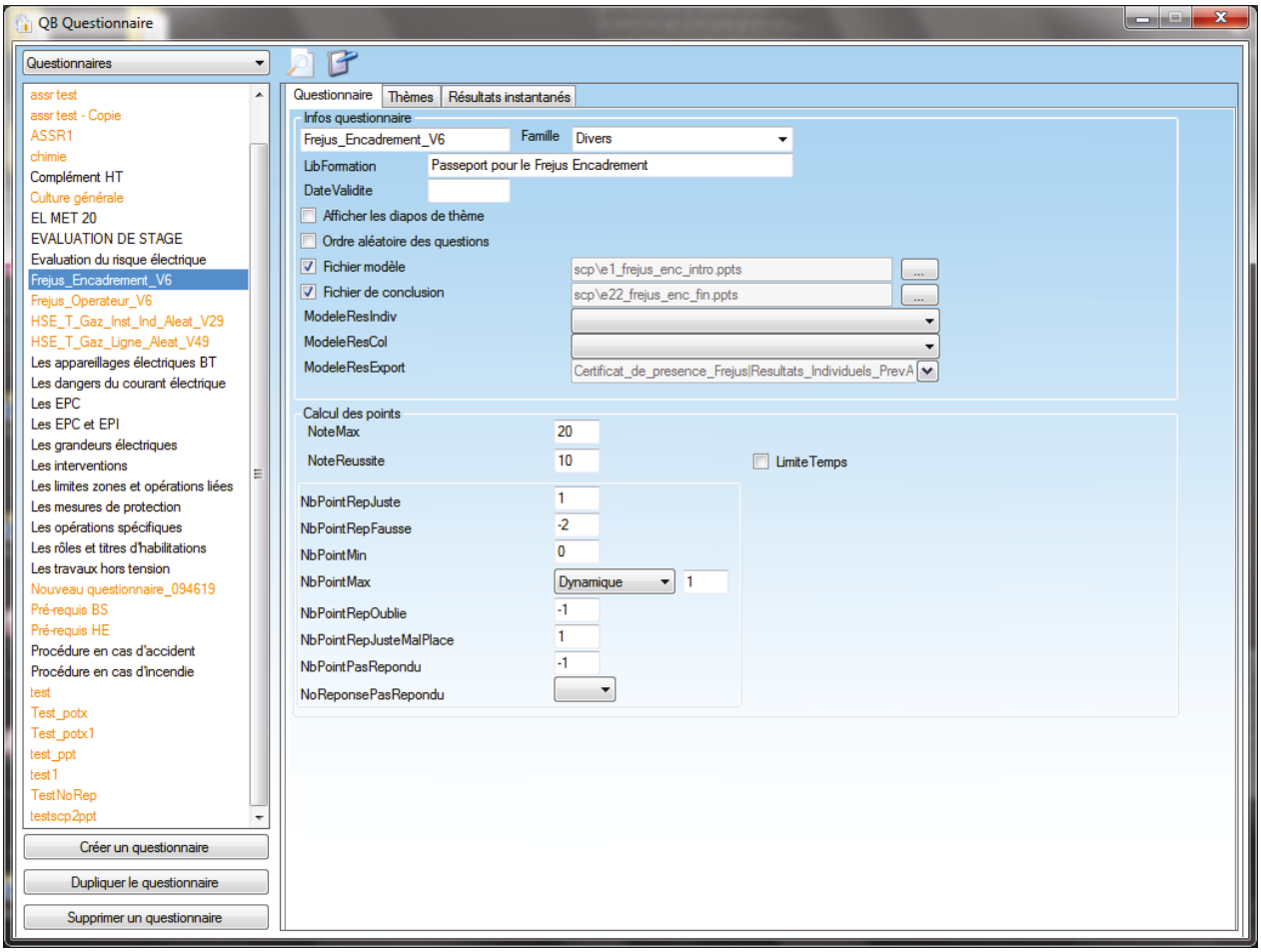


Fig. AN4.5 - Interface QB_Questionnaire

Principalement utilisé pour construire des questionnaires multi-sources, QB_Questionnaire est le programme de QuizzBox qui gère le référentiel des questionnaires publiés dans le système. La capture d'écran ci-dessus montre l'onglet de configuration des propriétés principales d'un questionnaire. L'onglet suivant (Thèmes) permet de choisir les différentes sources qui serviront à la construction du questionnaire, mais aussi le nombre de questions à sélectionner dans ces fichiers, de manière aléatoire ou pas. Le dernier onglet (Résultats instantanés) permet de choisir quels seront les modèles à utiliser lors de l'animation si on souhaite afficher en cours de questionnaire les résultats intermédiaires cumulés des stagiaires ou les résultats de la question en cours.

QB_Pratique

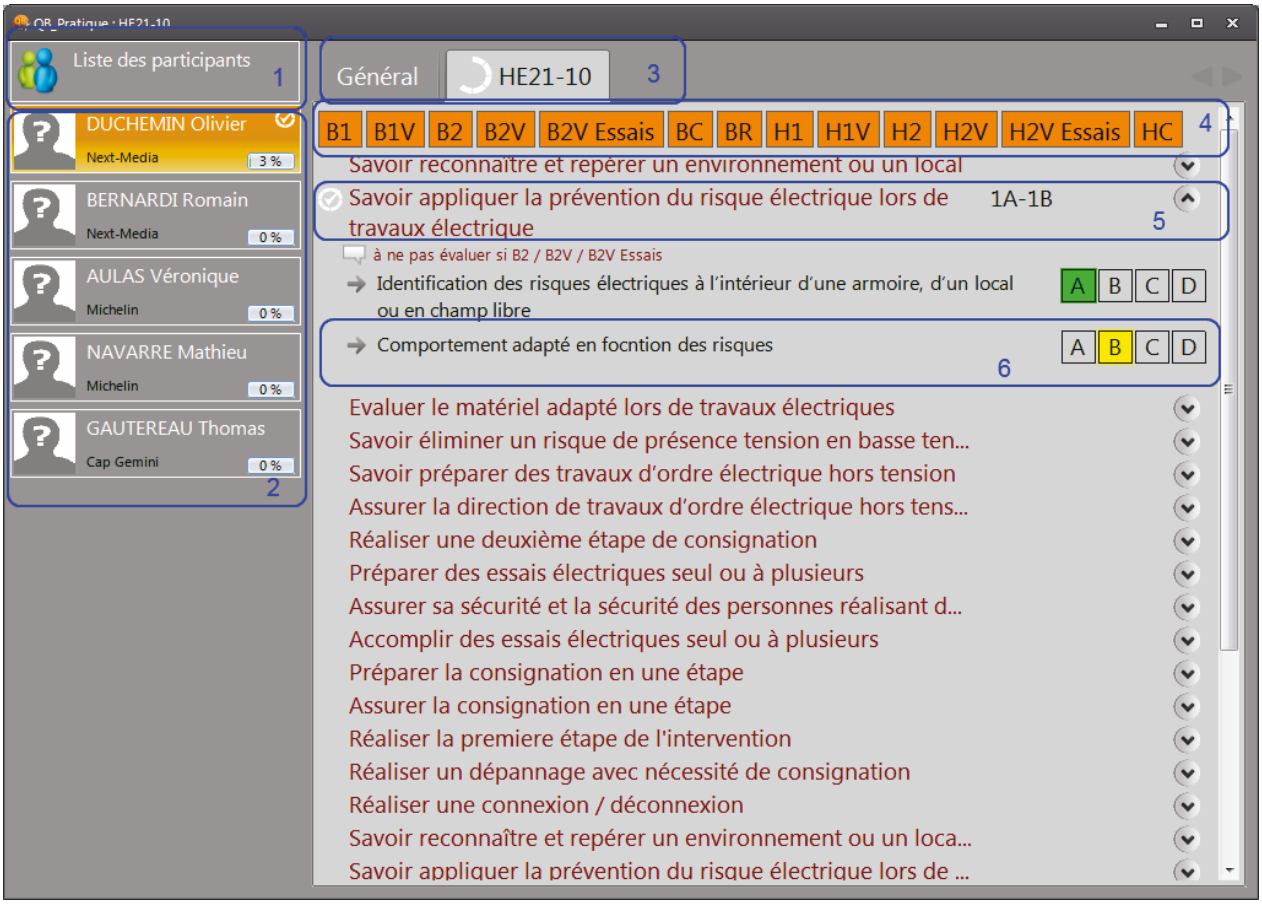


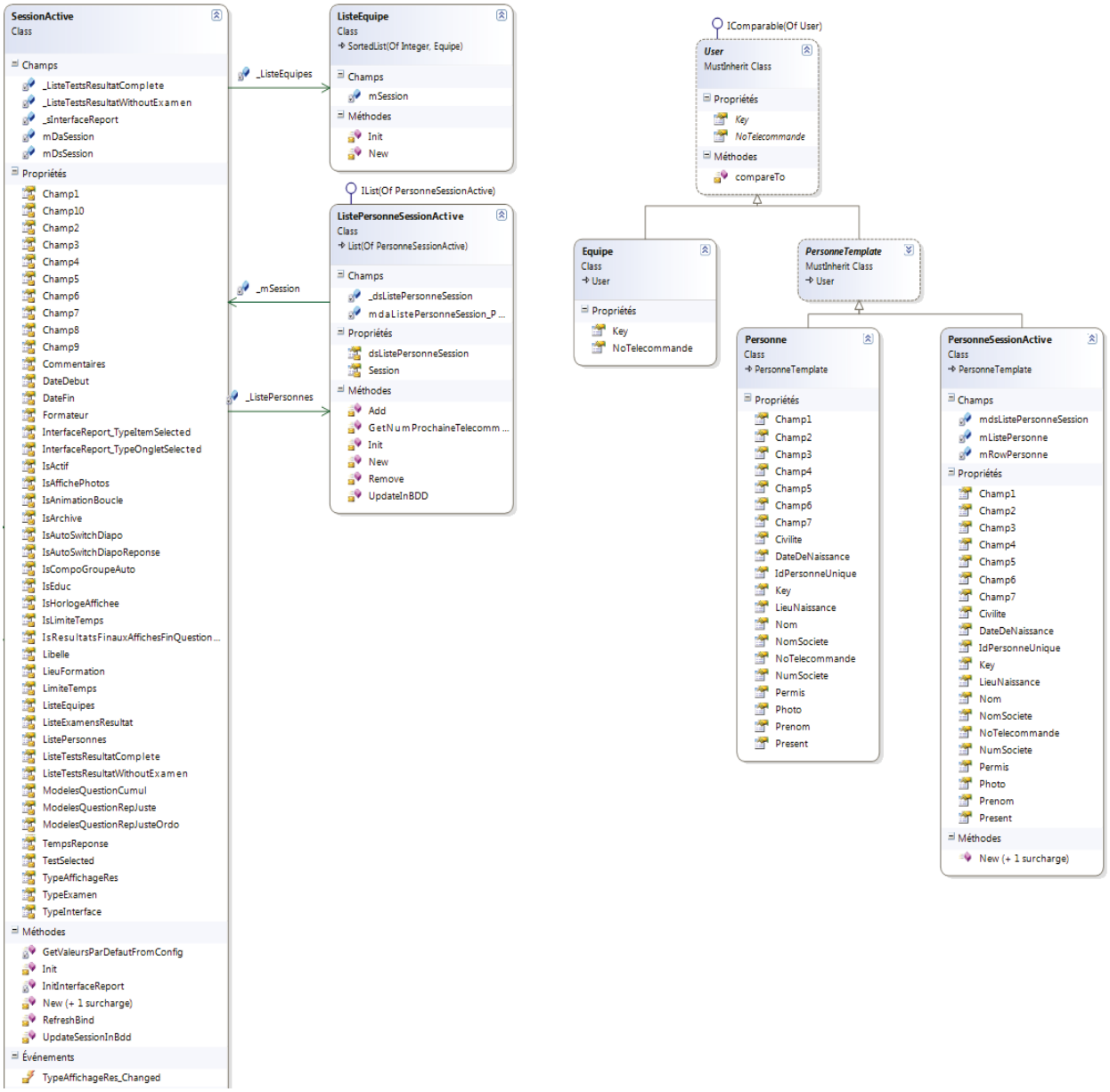
Fig. AN4.6 - Interface QB_Pratique

- Zone 1 : En cliquant sur la zone 1, QB_Pratique affiche dans la partie droite de son interface le récapitulatif des résultats des stagiaires, les habilitations obtenus, d'éventuels commentaires.
- Zone 2 : La zone 2 contient la liste des stagiaires inscrits à cette épreuve. Le formateur, en se référant à la capture d'écran est en cours d'évaluation du stagiaire Olivier Duchemin. Il en est à 3% des questions répondues de l'ensemble du test.
- Zone 3 : Le test est découpé en domaines affichés sous forme d'onglets. Dans cet exemple, la configuration du test pratique n'autorise pas ce type d'affichage qui est remplacé par un onglet unique du nom du test (HE21-10).

L'onglet général affiche un récapitulatif des résultats du stagiaire sélectionné dans la zone 2, celui qui est en cours d'évaluation.

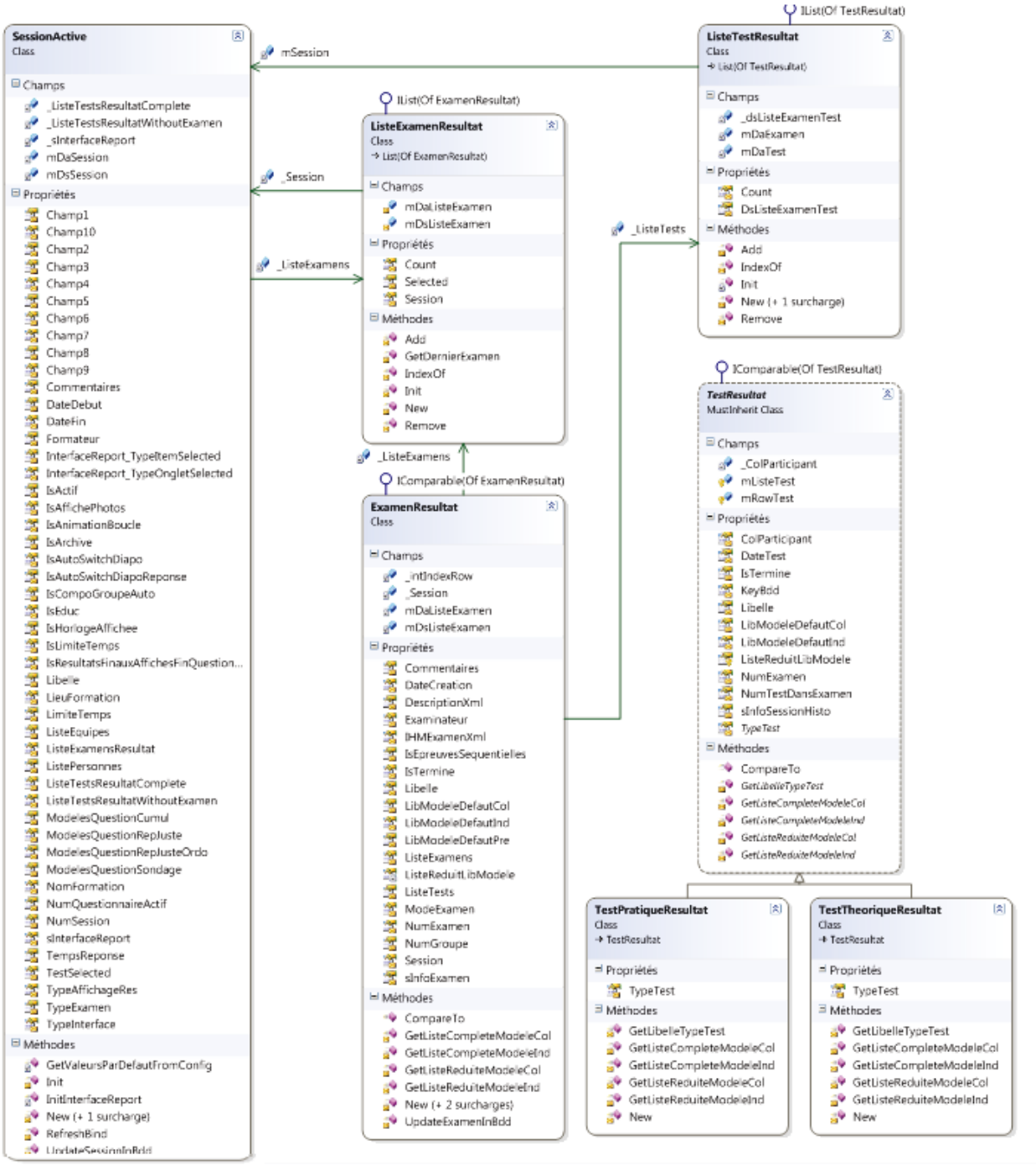
- Zone 4 : La zone 4 affiche et permet de filtrer sur les différents domaines où le stagiaire peut être évalué. Pour les domaines non concernés par l'évaluation, ceux-ci sont grisés
- Zone 5 : Il s'agit du thème en cours d'évaluation, l'indicateur de progression montre que le formateur a fini d'évaluer le stagiaire sur ce thème.
- Zone 6 : Point sur lequel le stagiaire doit être évalué. Le formateur a le choix entre 4 niveaux d'évaluation de A, la meilleure note à D la moins bonne. Il est possible dans le fichier XML de test de définir le nombre de niveaux d'évaluation, le type (alphabétique ou numérique) et même la couleur lors de sa sélection dans l'application.

ANNEXE 5 – Classes métier (Session/Personnes)

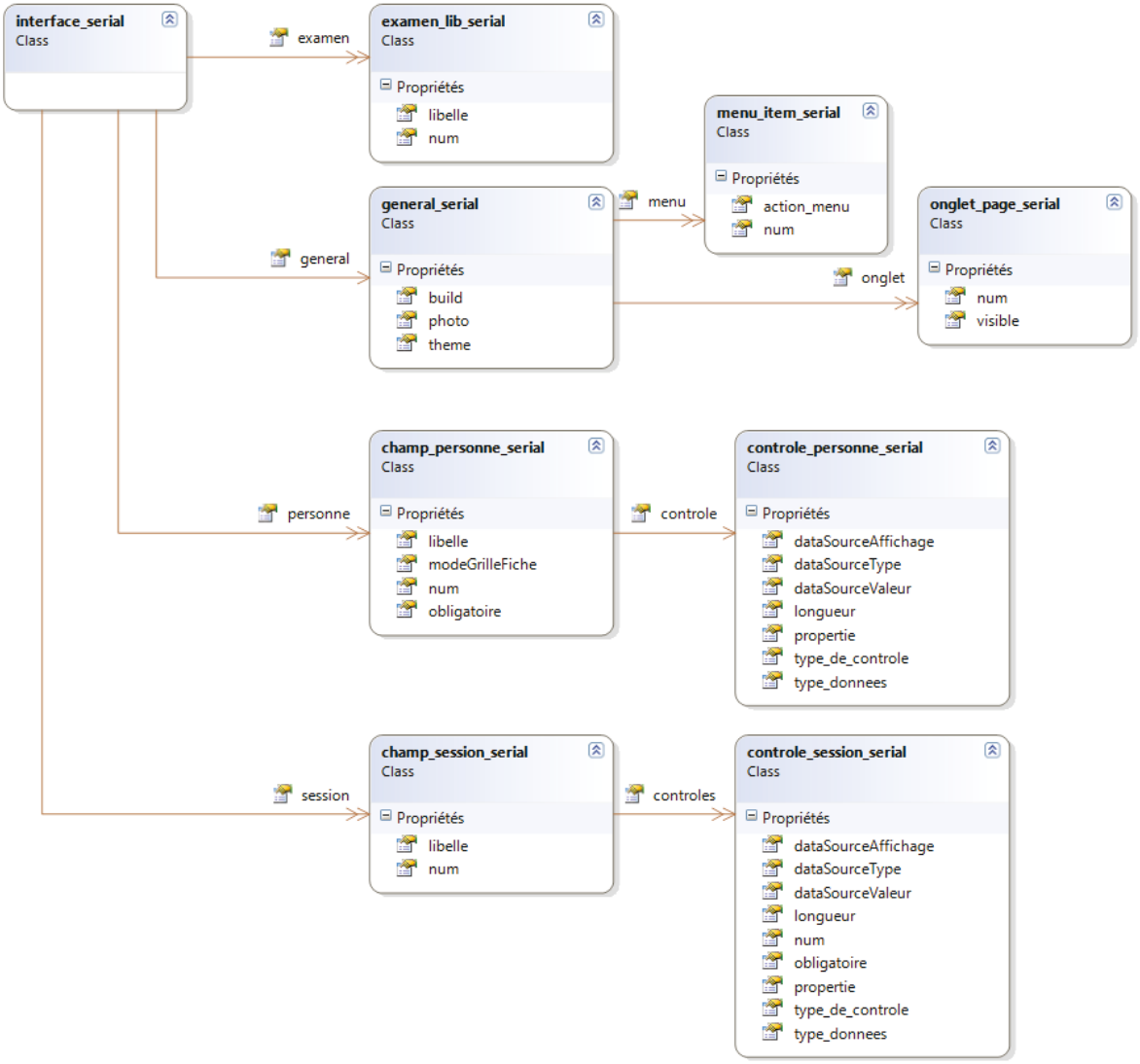


ANNEXE 6 – Classes métier (Tests)

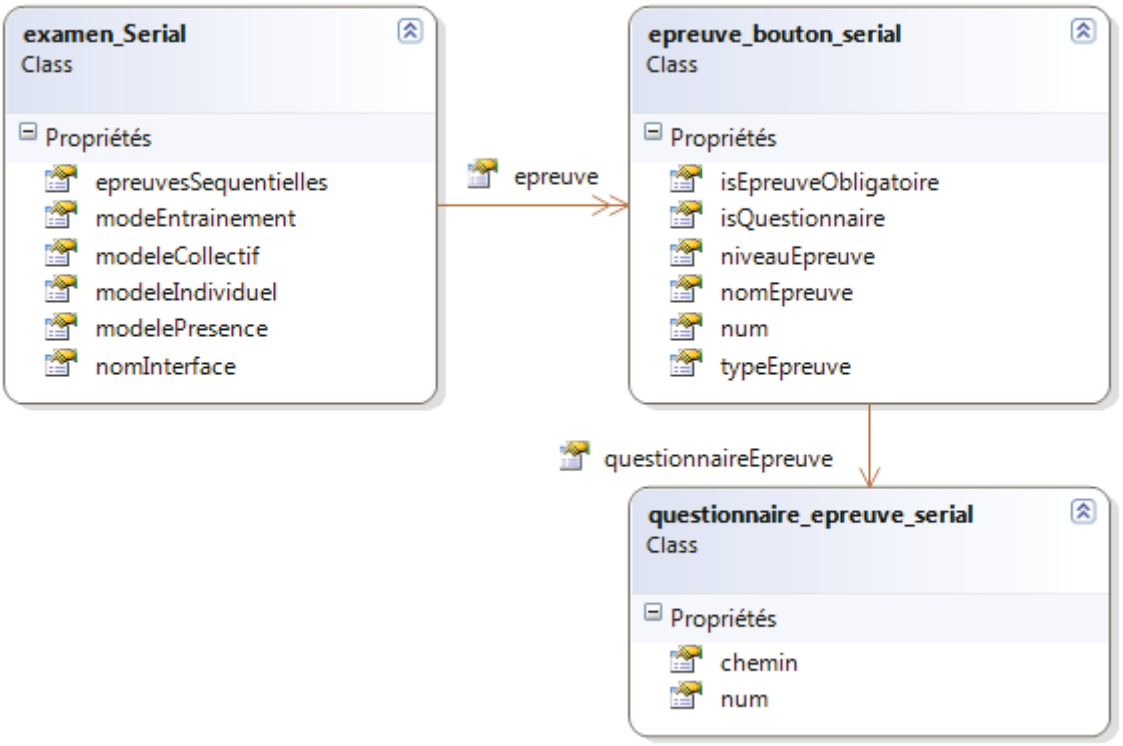
ANNEXE 7 – Classes métier (Résultats)



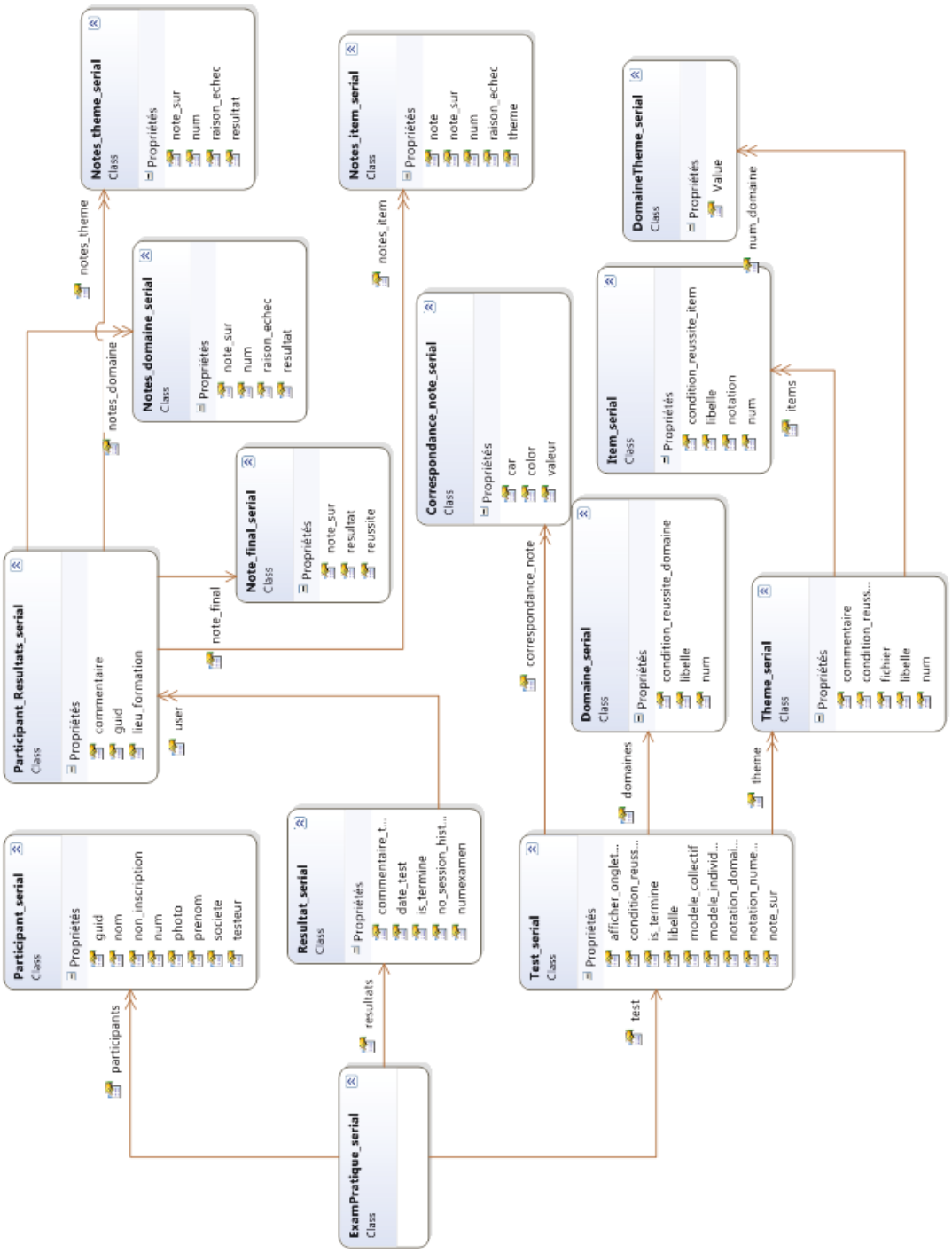
ANNEXE 8 – Classes de dé-sérialisation des interfaces



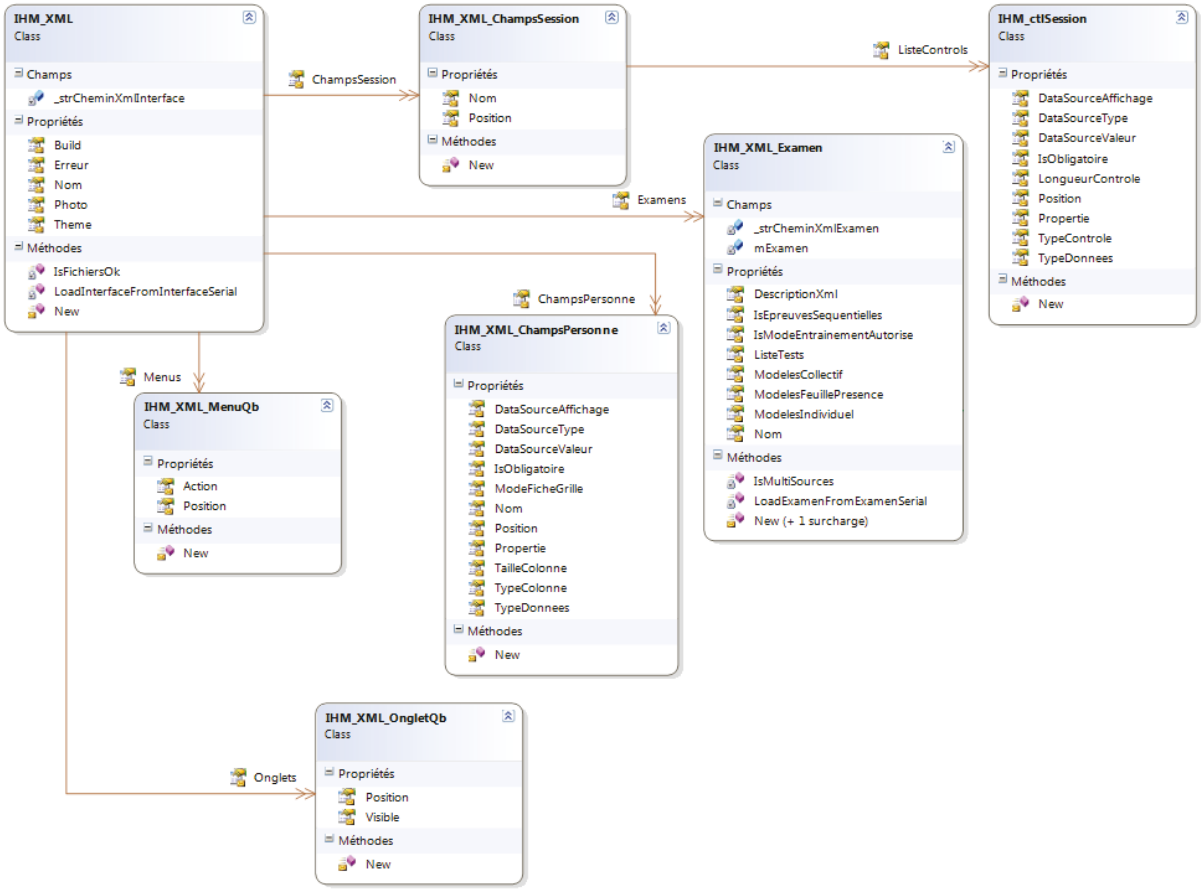
ANNEXE 9 – Classes de dé-sérialisation des examens



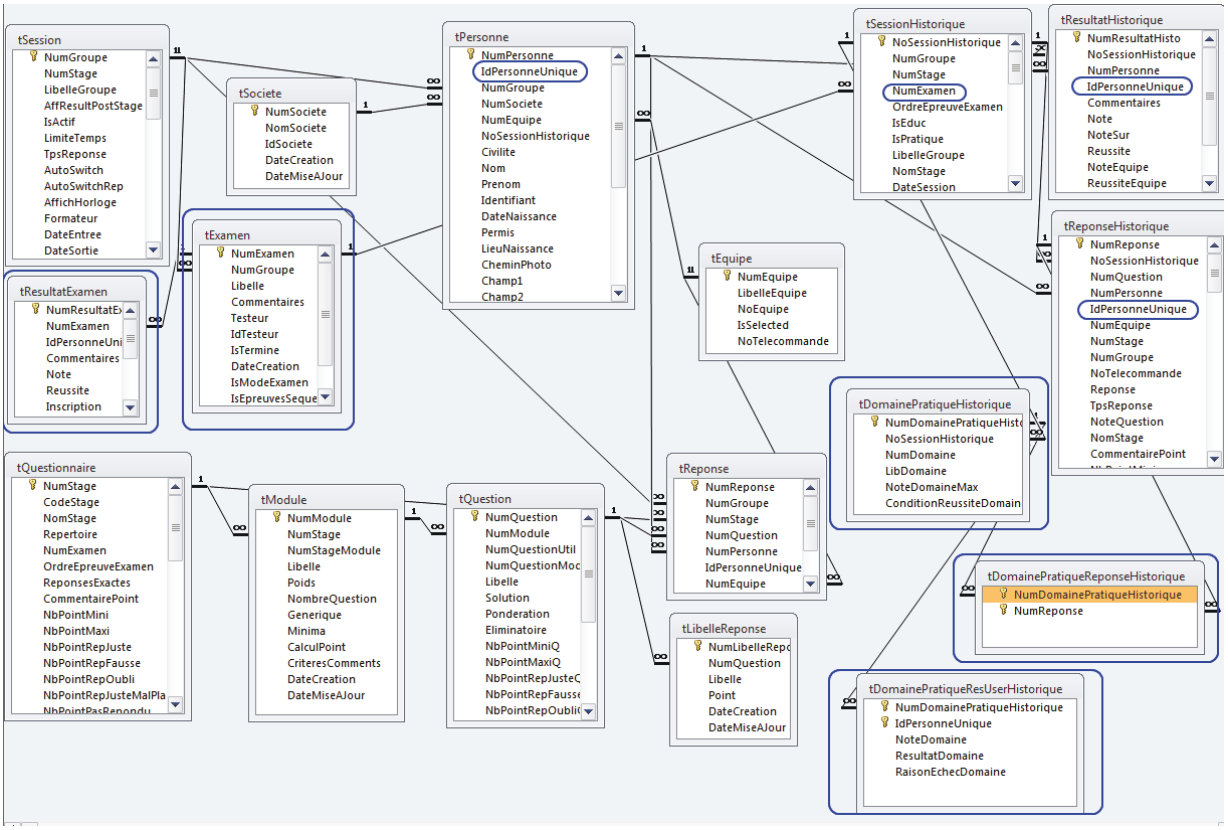
ANNEXE 10 – Classes de dé-sérialisation des épreuves pratiques



ANNEXE 11 – Classes de gestion des interfaces



ANNEXE 12 – Schéma de la base de données



Repéré en bleu sur le schéma ci-dessus, les évolutions principales apportées sur la version 5 à la base de données.

- Création de trois nouvelles tables (*tDomainePratiqueHistorique*, *tDomainePratiqueReponseHistorique* et *tDomainePratiqueResUserHistorique*) qui sont les tables accueillant les résultats des épreuves pratiques.
- Ajout du champ *IdPersonneUnique* à la table *tPersonne* pour la gestion de l'unicité des personnes dans le système par un GUID. Ce champ est répercuté dans les différentes tables de résultats.
- Ajout de la table *tExamen* et de sa clé primaire *NumExamen* propagée dans les différentes tables de résultats.
- Ajout de la table *tResultatExamen* qui stocke les notes des participants à la maille examen.

Bibliographie

Livres utilisés pendant le projet et la rédaction du mémoire

- T.GROUSSARD, *Visual Basic.net (VB.Net),* ENI, 2008, 510 pages
- B.A.GUERIN, *ADO.NET L'accès aux données,* ENI, 2010, 258 pages
- B.SHELDON, B.HOLLIS, K.SHARKEY, G.HILLAR, R.WINDSOR et J.MARBUTT, *Professional Visual Basic 2010 and .NET 4,* WROX PRESS, 2010, 1320 pages

Sites internet parcourus pendant le projet et la rédaction du mémoire

- Site communautaire tous langages de programmation : FAQ, tutoriaux, forums
  http://www.developpez.com
- Exemples de code Visual Basic
  http://www.vbfrance.com
- Introduction au .Net Remoting
  http://www.dotnet-tech.com/tutoriels/remoting/
- Msdn Microsoft
  http://msdn.microsoft.com
- Aginius
  http://www.aginius.fr

Liste des figures présentées dans le mémoire

Titre de la figure	Page
Figure 1 - Organigramme Next Media	12
Figure 2 - Système QB1	14
Figure 3 - Système QB2	14
Figure 4 - Exemple de question	17
Figure 5 - QB_Utilisateur, onglet "Résultats"	18
Figure 6 - Schéma des différents programmes de QuizzBox	20
Figure 7 - Les différents outils de la solution QuizzBox	23
Figure 8 - Schéma de principe de QB_Utilisateur	40
Figure 9 - Schéma de principe des interfaces	54
Figure 10 - Ecosystème de QB_Pratique	71

Refonte d'un système de vote interactif

Mémoire d'ingénieur CNAM, Auvergne 2012

RESUME

QuizzBox est un système de vote interactif conçu par la société Next Media. Il fonctionne à l'aide de boîtiers de vote, d'un récepteur USB et d'un PC équipé de Microsoft PowerPoint. Cette solution logicielle et matérielle permet de questionner, sonder une assemblée et recueillir les résultats.

Ce mémoire traite de la refonte de ce système. Elle a pour origine des besoins exprimés par un client ainsi qu'une nécessité pour l'équipe QuizzBox de faire évoluer sa suite logicielle.

Les principales évolutions apportées à QuizzBox sont l'intégration de la notion d'examen, l'ajout d'un module d'évaluation du savoir-faire ainsi que la personnalisation plus poussée de l'ensemble des outils.

Mots clés : QuizzBox, vote, interactif, examen, télécommande, questionnaire, stagiaire, formation

---

ABSTRACT

QuizzBox is an interactive voting system designed by the company Next Media. It works with remote controllers, an USB receiver and a computer with Microsoft PowerPoint. This software and material solution allows to question, to carry out an opinion, to poll and to collect the results.

This report deals with the revision of this system. It originates from the needs expressed by a customer and the necessity for the QuizzBox team to develop its software solution.

The main evolutions brought to QuizzBox are the integration of the concept of examination, the addition of an evaluation module of the know-how as well as a more elaborate customization for all the tools.

Keywords : QuizzBox, vote, interactive, exam, remote control, test, trainee, training

Vu le Professeur Principal,

M. SCHNEIDER

Vu le Directeur du Cnam d'Auvergne

P. CONIL