le cnam

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL DE LYON

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Informatique

OPTION : Système d'information

par

Cédric BARBE

Évolution des logiciels permettant de gérer l'alignement des électroaimants supraconducteurs du LHC

Soutenu le 16 Janvier 2014

JURY

PRESIDENT :	**M. PICOULEAU Christophe**	**CNAM PARIS**
MEMBRES :	**M. DAVID Bertrand**	**CNAM LYON**
	M. GENIER Claude	**CNAM LYON**
	M. MISSIAEN Dominique	**CERN**
	M. JONES Mark	**CERN**

Remerciements

En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de ces 8 années de cours au sein du CNAM.

Je tiens à remercier sincèrement Monsieur Dominique Missiaen, qui, en tant que responsable de la section SU au CERN, m'a permis de réaliser mon stage dans le cadre d'un congé individuel de formation. Je remercie mon entreprise TOMY France et en particulier Monsieur Lionel Delisle, Directeur Général, pour avoir validé sans hésiter ma demande de congé.

Mes remerciements s'adressent également à Monsieur Mark Jones, responsable de la sous section 'Accelerator and Computing', qui m'a accueilli à mon arrivé au CERN et qui a suivi le déroulement de mon travail, m'a intégré à l'équipe de développement et guidé dans les moments clés.

J'exprime ma gratitude au testeur chevronné de mon logiciel, Monsieur Michel Troillet, qui a pu trouver les bugs, même les plus indétectables, afin qu'il ait un comportement conforme à celui attendu et à Monique Tortrat, ma collègue de bureau avec qui j'ai passé d'excellents moments et qui a été à l'origine de mon stage au CERN.

Je tiens à remercier chaleureusement Catherine Bolgert pour son travail de relecture et de correction.

Enfin, j'adresse mes plus sincères remerciements à tous mes proches et amis, qui m'ont toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.

Glossaire

AD : Décélérateur d'antiprotons, machine unique en son genre, produit des antiprotons de basse énergie pour étudier l'antimatière, et en particulier créer des antiatomes.

ALICE : (A Large Ion Collider Experiment) est un détecteur d'ions lourds installé sur l'anneau du Grand collisionneur de hadrons (LHC). Il a été conçu pour étudier les propriétés physiques de la matière soumise à l'interaction forte, à des densités d'énergie extrêmes auxquelles une phase de la matière appelée plasma quarks-gluons se forme.

Android : Android, prononcé Androïd, est un système d'exploitation pour Smartphones, tablettes tactiles, PDA et terminaux mobiles. C'est un système open source utilisant le noyau Linux.

ApEx : Environnement de développement intégré permettant de créer des applications de type web en développement rapide et dont le but est d'exploiter des bases de données Oracle.

API : En informatique une interface de programmation (abr. API pour Application Programming Interface) est un ensemble normalisé de classes, des méthodes ou des fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur.

ATLAS : ATLAS est l'un des deux détecteurs polyvalents du Grand collisionneur de hadrons (LHC). Il étudie des domaines de physique très variés, de la recherche du boson de Higgs aux dimensions supplémentaires de l'espace-temps, en passant par les particules qui pourraient former la matière noire.

Blackberry OS : BlackBerry OS est un système d'exploitation propriétaire pour téléphone mobile de la gamme BlackBerry, conçu par la société canadienne Research In Motion (RIM), maintenant connue sous le nom de Blackberry.

C : Le C est un langage de programmation impératif, généraliste, issu de la programmation système. Inventé au début des années 1970 pour réécrire UNIX, C est devenu un des langages les plus utilisés.

C# : Le C# est un langage de programmation orienté objet à typage fort, créé par la société Microsoft, et notamment un de ses employés, Anders Hejlsberg, le créateur du langage Delphi.

C++ : Le C++ est un langage de programmation permettant la programmation sous de multiples paradigmes comme la programmation procédurale, la programmation orientée objet et la programmation générique.

CAST : CAST (l'acronyme de CERN Axion Solar Telescope) est une expérience en physique des particules, dont le but est de chercher des axions provenant du Soleil. L'expérience, placée au CERN en Suisse, est entrée en service en 2002 et la première détection fut effectuée en mai 2003. La détection réussie d'axions solaires devrait constituer une découverte majeure en physique des particules, et ouvrira de nouvelles portes quant au fonctionnement du noyau solaire.

CCS : (CERN Coordinate System) référentiel primaire au CERN sur lequel est défini tous les accélérateurs et installations.

CERN : L'Organisation européenne pour la recherche nucléaire, aussi appelée laboratoire européen pour la physique des particules et couramment désignée sous l'acronyme CERN (du nom du Conseil européen pour la recherche nucléaire, organe provisoire institué en 1952), est le plus grand centre de physique des particules du monde.

Chaba : CHABA est un programme employé pour déterminer les paramètres de transformation qui lient deux jeux de coordonnées Cartésiennes pour un groupe de points communs, chacun déterminé dans un système de référence local différent. Le programme fonctionne en mode 3D et la transformation est une transformation simple 3D Helmert.

CLIC : Le projet CLIC (Collisionneur linéaire compact), mené par une collaboration internationale, porte sur une machine où des électrons et des positons (antiélectrons) entreraient en collision à des énergies de plusieurs TeV. Le domaine d'énergie est similaire à celui du LHC, mais, en utilisant des électrons et leurs antiparticules au lieu de protons, les experts pensent ouvrir une perspective différente sur la physique sous-jacente.

CMS : Le Solénoïde compact pour muons (CMS) est un détecteur polyvalent installé sur l'anneau du LHC. Il a été conçu pour explorer un large éventail de domaines de la physique, allant de la recherche du boson de Higgs à celle d'autres dimensions, en passant par la quête des particules qui pourraient constituer la matière noire.

CNGS : Le projet Neutrinos du CERN vers le Gran Sasso (CNGS) consiste à envoyer des neutrinos muoniques depuis le CERN vers le laboratoire du Gran Sasso (LNGS) situé en Italie, à une distance de 732 kilomètres. Les neutrinos interagissent tellement peu avec les autres particules qu'ils passent aisément à travers la roche. Au Gran Sasso, les expériences OPERA et ICARUS tentent d'identifier les neutrinos tauiques issus des neutrinos muoniques. Le projet et terminé à ce jour.

COCCOMO : COCOMO (acronyme de l'anglais COnstructive COst MOdel) est un modèle permettant de définir une estimation de l'effort à fournir dans un développement logiciel et la durée que ce dernier prendra en fonction des ressources allouées.

Compass : COMPASS est une expérience polyvalente qui fait partie du programme du Supersynchrotron à protons du CERN. Elle étudie les liens complexes qui se tissent entre les quarks et les gluons et qui leur permettent de donner naissance aux particules observables, depuis le proton jusqu'à la vaste palette des particules plus complexes.

CSS : CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML et XML.

Emq : Erreur quadratique moyenne - Mesure d'écart de prédiction par rapport à la valeur réelle (calculée comme la racine carrée de l'écart quadratique moyen).

ENT : Méthode des points de fonction - Entrée : fonctions d'entrée de données métier de l'utilisateur dans le périmètre du système. Créations, modifications, duplications, validations, suppressions.

eV : En physique, l'électron-volt1 (symbole eV) est une unité de mesure d'énergie. Sa valeur est définie comme étant l'énergie acquise par un électron accéléré depuis le repos par une différence de potentiel d'un volt : 1 eV = e•(1 V) (ou en forme courte « eV = e•V »), où e est la valeur absolue de la charge de l'électron (c'est-à-dire la charge électrique dite élémentaire). Un électron-volt est donc égal à environ 1,602176565×10-19 joule (J)2. C'est une unité hors système international (SI) dont la valeur est obtenue expérimentalement.

FA : Méthode des points de fonction : facteur d'ajustement

GDE : Méthode des points de fonction - Groupe de données externes : Groupe de données liées logiquement, ou de paramètre de contrôle, identifiables par l'utilisateur, utilisés par l'application, mis à jour par une autre application.

GDI : Méthode des points de fonction - Groupe de données internes : Groupe de données liées logiquement, ou de paramètres de contrôle identifiables par l'utilisateur, mis à jour et utilisés à l'intérieur de la frontière de l'application.

Géode : Géode est l'interface utilisateur permettant l'accès à la base de données Survey. Elle est destinée principalement au personnel de la section SU.

GeV : 1 GeV = 10 exp 9 eV

GPS : Le Global Positioning System (GPS) – que l'on peut traduire en français par « système de localisation mondial » – est un système de géolocalisation fonctionnant au niveau mondial. En 2011, il est avec GLONASS, un système de positionnement par satellites entièrement opérationnel et accessible au grand public.

GUI : En informatique, une interface graphique (anglais GUI pour graphical user interface) est un dispositif de dialogue homme-machine, dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, que l'usager peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage, le plus souvent une souris.

INT : Méthode des points de fonction - Interrogation : fonctions de consultation des données métiers. Listes, détails, annotations.

iOS : iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch et l'iPad.

ISOLDE : L'installation ISOLDE, unique en son genre, produit des faisceaux de basse énergie de nucléides radioactifs, dont l'instabilité est due soit à un excédent, soit à un déficit de neutrons. En fait, l'installation réalise le vieux rêve des alchimistes : la transmutation d'un élément en un autre. Elle permet ainsi l'étude des noyaux atomiques, y compris les espèces les plus exotiques.

J2ME : Java Platform, Micro Edition ou Java ME (anciennement J2ME) est le Framework Java spécialisé dans les applications embarquées (lecteurs blu-ray, téléphonie mobile, automobile, domotique, électroménager...). Il est né en 1999, de la volonté de Sun de structurer ses différentes plateformes, et notamment Java Card, Personnal Java et Embedded Java sous l'appellation J2ME.

Java : Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de

Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

JavaScript : JavaScript (souvent abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.

jQuery : jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.

JVM : La machine virtuelle Java (en anglais Java virtual machine, abr. JVM) est un appareil informatique fictif qui exécute des programmes compilés sous forme de bytecode Java. Architecture générale : illustration du slogan Compile once, run everywhere. L'appareil est simulé par un logiciel spécifique à chaque plate-forme ou couple (machine/système d'exploitation) et permet aux applications Java compilées en bytecode de produire les mêmes résultats quelle que soit la plate-forme, tant que celle-ci est pourvue de la machine virtuelle Java adéquate.

LEIR : L'Anneau d'ions de basse énergie (Low Energy Ion Ring - LEIR) reçoit de l'accélérateur linéaire 3 (Linac 3) de longues impulsions d'ions plomb qu'il transforme en paquets courts et denses, qui seront injectés dans le Grand collisionneur de hadrons(LHC).

LGC : LGC (Logiciel Général de Compensation) est un outil pour le traitement des réseaux géodésiques du CERN. Le programme permet un traitement rigoureux de réseaux tridimensionnels.

LHC : Le Large Hadron Collider (LHC, ou Grand collisionneur de hadrons1 en français) est un accélérateur de particules mis en fonctionnement le 10 septembre 2008 et inauguré officiellement le 21 octobre 2008 au CERN. Situé à la frontière franco-suisse, c'est le plus puissant accélérateur de particules au monde construit à ce jour. Il est même présenté comme le plus grand dispositif expérimental jamais construit pour valider des théories physiques.

LHCb : L'expérience LHCb (Large Hadron Collider beauty) explore les légères différences qui existent entre matière et antimatière grâce à l'étude d'un type de particule appelé « quark beauté » ou « quark b ».

LHCf : L'expérience LHCf (Large Hadron Collider forward) utilise les particules à petits angles créées lors des collisions produites dans le Grand collisionneur de hadrons (LHC) pour simuler des rayons cosmiques en conditions de laboratoire. Les rayons cosmiques sont des particules chargées provenant de l'espace interstellaire, qui bombardent constamment l'atmosphère terrestre. Arrivés dans la haute atmosphère, ils percutent des noyaux d'atomes, ce qui produit une cascade de particules au sol. L'étude de la façon dont les collisions dans le LHC produisent des cascades similaires pourrait aider les physiciens à étalonner les détecteurs des gigantesques expériences sur les rayons cosmiques (certaines peuvent couvrir des milliers de kilomètres), ainsi qu'à interpréter leurs résultats.

Linac 4 : L'accélérateur linéaire 4 (Linac 4) est conçu pour porter des ions d'hydrogène négatifs à des énergies élevées. Il constituera la source de faisceaux de protons du Grand collisionneur de hadrons (LHC) après le long arrêt prévu en 2017-2018.

MeV : 1 MeV = 10 exp 6 eV

MoEDAL : En 2010, le Grand collisionneur de hadrons (LHC) a approuvé sa septième expérience : MoEDAL (Monopole and Exotics Detector at the LHC) MoEDAL a pour principale mission de rechercher directement le monopôle magnétique, une particule hypothétique portant une charge magnétique.

MVC : Le patron modèle-vue-contrôleur (en abrégé MVC, de l'anglais model-view-controller), tout comme les patrons modèle-vue-présentation ou Présentation, abstraction, contrôle, est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective.

nTOF : L'installation de mesure du temps de vol des neutrons (nTOF) fonctionne au CERN depuis 2001. Il s'agit d'une source de neutrons pulsés associée à une trajectoire de 200 mètres qui a été conçue pour étudier les interactions noyau - neutron pour des énergies de neutron allant de quelques MeV à plusieurs GeV. Cette vaste gamme d'énergie et les faisceaux de neutrons d'intensité élevée produits à nTOF sont utilisés pour effectuer des mesures précises des processus liés aux neutrons.

Objective-C : L'Objective-C est un langage de programmation orienté objet réflexif. C'est une extension du C ANSI, comme le C++, mais qui se distingue de ce dernier par sa distribution dynamique des messages, son typage faible ou fort, son typage dynamique et son chargement dynamique. Contrairement au C++, il ne permet pas l'héritage multiple mais il existe toutefois des moyens de combiner les avantages de C++ et d'Objective-C.

PFA : Méthode des points de fonction : points de fonction ajustés.

PFB : Méthode des points de fonction : points de fonction bruts.

PFB : Pocket Field Book. Logiciel sur Pocket PC permettant différentes prises de mesures.

PocketPc : Microsoft Pocket PC est le nom des OS (ou systèmes d'exploitation) de Microsoft pour assistants personnels. Microsoft Pocket PC est l'évolution des versions Windows CE.

PS : Elément clé du complexe d'accélérateurs, le Synchrotron à protons (PS) accélère les protons fournis par le Booster (PSB) ou les ions lourds provenant de l'Anneau d'ions de basse énergie (LEIR). Tout au long de son histoire, le PS a jonglé avec des particules de toutes sortes, soit pour injecter les faisceaux directement dans les expériences, soit pour alimenter des accélérateurs plus puissants.

PSB : Le Synchrotron à Protons Booster est composé de quatre anneaux synchrotron superposées qui reçoivent des faisceaux de protons de l'accélérateur linéaire Linac 2 à 50 MeV et accélèrent les 1,4 GeV pour l'injection dans le Synchrotron à protons (PS).

QML : QML (Qt Meta Language ou Qt Modeling Language) est un langage déclaratif basé sur JavaScript pour la conception d'applications d'interface centrée sur l'utilisateur. Il fait partie de Qt Quick, le kit de création d'interface utilisateur développée par Nokia dans le cadre Qt. QML est principalement utilisé pour les applications mobiles où entrée tactile, des animations fluides (60 FPS) et l'expérience utilisateur sont cruciales.

Qt : Qt (prononcé officiellement en anglais cute est une API orientée objet et développée en C++ par Qt Development Frameworks, filiale de Digia. Qt offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc.

QWidget : La classe QWidget est la classe de base de tous les objets d'interface utilisateur.

RS232 : RS-232 (parfois appelée EIA RS-232, EIA 232 ou TIA 232) est une norme standardisant un bus de communication de type série sur trois fils minimum (électrique,

mécanique et protocole). Disponible sur presque tous les PC jusqu'au milieu des années 2000, il est communément appelé le « port série ». Sur les systèmes d'exploitation MS-DOS et Windows, les ports RS-232 sont désignés par les noms COM1, COM2, etc. Cela leur a valu le surnom de « ports COM », encore utilisé de nos jours. Cependant, il est de plus en plus remplacé par le port USB.

SOR : Méthode des points de fonction - Sortie : fonctions de restitution des données transformées. Calculs, graphiques, déductions.

SPS : Avec sa circonférence de 7 km, le Supersynchrotron à protons est la plus grande machine du complexe d'accélérateurs après le Grand collisionneur de hadrons (LHC). Il reçoit ses particules du synchrotron à protons (PS) et les accélère pour alimenter en faisceaux le LHC, les expériences NA61/SHINE et NA62, l'expérience COMPASS et le projet CNGS.

SU : La section SU (Large Scale métrologie) est responsable de la métrologie et de l'alignement des accélérateurs, de leurs lignes de transfert de faisceau associés et des détecteurs, pour l'ensemble du site du CERN.

TeV : 1 TeV = 10 exp 12 eV

TOTEM : L'expérience TOTEM (TOTal Elastic and diffractive cross section Measurement) étudie des particules à très petits angles, une partie de la physique inaccessible aux expériences polyvalentes. Entre autres recherches, TOTEM va par exemple mesurer la taille des protons et évaluer précisément la luminosité du LHC.

UI : (User Interface). Les interactions homme-machine (IHM) définissent les moyens et outils mis en œuvre afin qu'un humain puisse contrôler et communiquer avec une machine. Les ingénieurs en ce domaine étudient la façon dont les humains interagissent avec les ordinateurs ou entre eux à l'aide d'ordinateurs, ainsi que la façon de concevoir des systèmes qui soient ergonomiques, efficaces, faciles à utiliser ou plus généralement adaptés à leur contexte d'utilisation.

UML : En informatique UML (de l'anglais Unified Modeling Language), ou Langage de modélisation unifié, est un langage de modélisation graphique à base de pictogrammes. Il est utilisé en développement logiciel, et en conception orientée objet. UML est couramment utilisé dans les projets logiciels.

VB.Net : Visual Basic .Net est un langage de programmation à la syntaxe similaire à celle de Visual Basic 6. Néanmoins, ces deux langages sont assez peu comparables dans la pratique tant l'évolution entre ceux-ci est énorme. Le principal changement étant sans conteste l'introduction de l'orientation objet dans le langage.

Web : Le World Wide Web (WWW), littéralement la « toile (d'araignée) mondiale », communément appelé le Web, et parfois la Toile, est un système hypertexte public fonctionnant sur Internet. Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles

WebApp : En informatique, une application web (aussi appelée web app, de l'anglais) est une application manipulable grâce à un navigateur web. De la même manière que les sites web, une application web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.).

Windows Phone : Windows Phone est un système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, sa précédente plateforme logicielle qui a été renommée pour l'occasion en Windows Phone Classic1'2. Contrairement au système qu'il remplace, Windows Phone est d'abord principalement destiné au grand public plutôt qu'au marché des entreprises3.

WLCG : Le "Worldwide LHC Computing Grid" (la grille de calcul) est un vaste réseau mondial de centres de calcul qui transmettent, stockent, traitent et analysent les données du Large Hadron Collider en continu.

WPS : Wire Positioning System. Système de positionnement par fil tendu pour réaliser l'alignement de différentes machines.

Table des matières

Remerciements 3
Glossaire 4
Table des matières 14
Introduction 17
I LE CERN 19
I.1 PRESENTATION 19
I.1.1 LE NOM 19
I.1.2 LES ETATS MEMBRES (CERN, 2013) 19
I.2 LA STRUCTURE DU CERN (CERN, 2013) 20
I.2.1 LE CONSEIL 21
I.2.2 LE COMITE DES DIRECTIVES SCIENTIFIQUES 21
I.2.3 LE COMITE DES FINANCES 21
I.2.4 LE DIRECTEUR GENERAL 21
I.2.5 LES DEPARTEMENTS 21
I.3 LA PHYSIQUE (CERN, 2013) 22
I.4 LES INSTALLATIONS 23
I.4.1 LES ACCELERATEURS (CERN, 2013) 24
I.4.2 LES EXPERIENCES (CERN, 2013) 25
I.5 LE SERVICE INFORMATIQUE 26
I.6 LA SECTION SU 27
I.6.1 PRESENTATION 27
I.6.2 LE SERVICE INFORMATIQUE DE LA SECTION 28
I.6.3 LE FONCTIONNEMENT DE LA SECTION SU 29
I.6.4 LE SYSTEME METROLOGIQUE DU CERN 30
I.6.4.1 Le système de coordonnée 30
I.6.4.2 Le système géodésique de référence 30
I.6.4.3 Autres systèmes de référence (Mark Jones, 2005) 32
I.7 LE STOCKAGE DANS LA BASE DE DONNEES SURVEY 32
I.8 LES DIFFERENTS TYPES DE MESURES 32
I.8.1 LE NIVELLEMENT 33
I.8.2 L'OFFSET 34
I.8.3 LE TILT 34
I.8.4 AUTRES MESURES 35
II DESCRIPTION DU PROJET 37
II.1 CAHIER DES CHARGES 37
II.2 SPECIFICITE DU DEVELOPPEMENT 38
II.2.1 DOCUMENTATION EXISTANTE 38
II.2.2 SEPARATION ENTRE LES DONNEES ET LA VUE (GUI) 38
III ANALYSE DE L'EXISTANT 39
III.1 LE LOGICIEL PFB 39
III.2 L'INTERFACE GRAPHIQUE 39
III.3 LE DATA WORKFLOW 45
IV CHOIX DE LA PLATEFORME DE DEVELOPPEMENT 47
IV.1 LES DIFFERENTES SOLUTIONS EXISTANTES 47

IV.1.1 L'APPROCHE WEBAPP 47
IV.1.1.1 jQuery Mobile 47
IV.1.1.2 Sencha Touch 47
IV.1.1.3 Titanium 48
IV.1.1.4 PhoneGap 48
IV.1.1.5 Rhodes 48
IV.1.2 LA SOLUTION 'APPLICATION NATIVE' 49
IV.1.2.1 Qt 49
IV.1.2.2 Java (Virtual Machine sur OS Mobile) 49
IV.1.2.3 MoSync SDK (MoSync AB, 2013) 50
IV.1.2.4 Haxe (Haxe, 2013) 50
IV.2 CHOIX DE LA SOLUTION 51
IV.3 LE FRAMEWORK QT (DIGIA OYJ, 2013) 52
IV.3.1 LES SIGNAUX ET LES SLOTS 54
IV.3.2 L'ARCHITECTURE MVC 54
IV.3.3 QT CREATOR ET QT DESIGNER 55
V CONCEPTION ET IMPLEMENTATION 57
V.1 INITIALISATION SOUS FORME DE GENIE LOGICIEL 57
V.2 ESTIMATION DE L'EFFORT A PRODUIRE POUR LE DEVELOPPEMENT 57
V.2.1 METHODE DES POINTS DE FONCTION 57
V.2.1.1 Dénombrements des différents composants 57
V.2.1.2 Calcul des points de fonction brut 57
V.2.1.3 Calcul des points de fonction ajusté 58
V.2.1.4 Calcul de la charge de travail 58
V.2.2 METHODE COCCOMO 58
V.3 MISE EN PLACE D'UN PLANNING DE DEVELOPPEMENT 59
V.3.1 UTILISATION DE JIRA 60
V.4 POLITIQUE DE DEVELOPPEMENT PAR MODULE 60
V.5 LA CONCEPTION GENERALE ET DETAILLEE 60
V.5.1 LES DIAGRAMMES D'ACTIVITE – MODULE DE TILT 61
V.5.2 LA CONCEPTION DETAILLEE – MODULE DE TILT 62
V.5.2.1 Diagramme de séquences – Validation des références 62
V.5.3 LES DIAGRAMMES D'ACTIVITE – MODULE DE NIVELLEMENT 63
V.5.4 LA CONCEPTION DETAILLEE – MODULE DE NIVELLEMENT 64
V.5.4.1 Diagramme de séquences – Sélection des points 64
V.5.4.2 Diagramme de séquences – Ecriture et enregistrement des mesures 65
V.5.5 LES DIAGRAMMES D'ACTIVITE – MODULE D'ECARTOMETRIE 67
V.5.6 LA CONCEPTION DETAILLEE – MODULE D'ECARTOMETRIE 68
V.5.6.1 Diagramme de séquences – Sélection d'une séquence 68
V.6 LA REECRITURE DU LOGICIEL 68
V.6.1 DIFFICULTES 69
V.6.1.1 Accès au système de fichier 69
V.6.1.2 Adaptation de l'interface graphique à un affichage tactile 70
V.6.1.3 Développement des classes permettant les mesures en automatique 71
V.6.2 L'ADAPTATION DE L'INTERFACE GRAPHIQUE A DIFFERENT TERMINAL 72
V.6.3 LES TESTS 72
V.6.4 LES RETOURS UTILISATEURS ET CORRECTIONS REALISEES 73
V.6.4.1 Corrections réalisées 73
V.6.4.2 Optimisations réalisées 74
V.6.5 BILAN DE LA REECRITURE 76
VI MAINTENANCE ET EVOLUTION 77

VI.1 LA DOCUMENTATION DESTINEE AU DEVELOPPEUR 77
VI.2 LA DOCUMENTATION DESTINEE AUX UTILISATEURS 77
Conclusion 78
Bibliographie 79
Table des annexes 82
Annexe 1 Structure du CERN 83
Annexe 2 Organigramme du département BE 84
Annexe 3 Structure du département ABP 85
Annexe 4 Structure de la section SU 87
Annexe 5 Qt Designer – Ecran d'édition des fenêtres 88
Annexe 6 Points de fonction – Dénombrement des composants 89
Annexe 7 Points de fonction – Addition des points de fonction 91
Annexe 8 Planning prévisionnel de développement – Diagramme de Gantt 92
Annexe 9 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Tilt 93
Annexe 10 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Nivellement 94
Annexe 11 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Ecartométrie 95
Annexe 12 Diagramme d'activité – Module de tilt 96
Annexe 13 Diagramme de séquence – Module de tilt – Validation des références 99
Annexe 14 Diagramme d'activité – Module de nivellement 100
Annexe 15 Diagramme de séquence – Module de nivellement – Sélection des points 105
Annexe 16 Diagramme de séquence – Module de nivellement – Ecriture et enregistrement des mesures 106
Annexe 17 Diagramme d'activité – Module d'écartométrie 112
Annexe 18 Diagramme de séquence – Module d'écartométrie – Sélection d'une séquence 116
Annexe 19 Free Serial Port Monitor 118
Annexe 20 Utilisation des layouts sous Qt Designer 119
Annexe 21 Tests d'intégration continus : comparaison des deux versions de PFB 121
Annexe 22 Correction du problème de multiples enregistrements 122
Annexe 23 Nivellement - Affichage du journal 123
Annexe 24 Documentation générée par Doxygen 124
Annexe 25 Documentation utilisateur sur Confluence 125
Liste des tableaux 127

Introduction

Le CERN, l'Organisation Européenne pour la Recherche Nucléaire est l'un des plus grands laboratoires de recherche sur la physique des particules dans le monde. Il a pour vocation la physique fondamentale et l'étude de la création de l'univers en étudiant la matière, son origine, et les forces qui la maintiennent agglomérée.

Pour étudier la matière, le CERN crée des faisceaux de particules qui circulent dans un accélérateur à des vitesses très élevées, de l'ordre de la vitesse de la lumière. Les faisceaux ainsi accélérés entrent en collision avec d'autres faisceaux ou avec des cibles fixes. Les détecteurs présents sur les lieux des collisions observent et enregistrent le résultat des collisions.

Afin que les faisceaux soient de la meilleure qualité possible, il est impératif d'être le plus précis possible dans le positionnement des différents éléments composant les accélérateurs et les détecteurs. C'est le rôle de la section SU (Large Scale Metrologie) où j'ai réalisé mon stage, qui, en relation avec les physiciens, veille à la bonne qualité des faisceaux grâce à un alignement très précis des éléments.

Ainsi, la section SU assure la métrologie des accélérateurs et des expériences sur 63 km cumulés de lignes de faisceaux et pour la plupart des expériences. Les données recueillies lors des différentes mesures sont insérées dans la base de données nommée Survey et sont exploitées grâce à une application associée nommée Géode et sont maintenues depuis plusieurs années par la section.

La mission que l'on m'a confiée à consister à la réécriture d'un logiciel nommé PFB (pour Portable Field Book) développé dans la section avec le langage VB.Net. Ce logiciel assiste les géomètres dans leurs prises de mesures et utilise pour cela un assistant électronique (PocketPc) fonctionnant sous Windows Mobile.

Ce type d'appareil mobile est aujourd'hui obsolète et a été supplanté par les quatre systèmes d'exploitation majeurs utilisés dans les appareils mobiles aujourd'hui : Blackberry OS, iOS, Android et Windows Phone.

On m'a demandé deux choses principales pour la réécriture du logiciel : que celui-ci soit multiplateforme (donc qu'il puisse au minimum fonctionner sur iOS et Android les deux principaux systèmes d'exploitation mobiles à ce jour), et de séparer la partie 'données' du code de la partie 'interface graphique'.

Je présenterai dans une première partie le CERN et la section dans laquelle j'ai été accueilli en expliquant les différentes mesures réalisées sur le terrain.

Dans la deuxième partie sera présenté le cahier des charges et les contraintes du projet. La partie suivante détaillera l'analyse du logiciel existant.

La quatrième partie présentera la phase conception et implémentation du nouveau logiciel tandis que la dernière détaillera la documentation développeur et utilisateur pour aider à l'utilisation et la maintenance future de l'application.

Je terminerai par le bilan de ce stage au sein du CERN.

I Le CERN

I.1 Présentation

I.1.1 Le nom



Le CERN est aujourd'hui « l'Organisation européenne pour la Recherche nucléaire ». À l'origine, l'acronyme correspondait à « Conseil européen pour la Recherche nucléaire », un organe provisoire institué en 1952, qui avait pour mandat de créer en Europe une organisation de rang mondial pour la recherche en physique fondamentale. Á l'époque, la recherche en physique fondamentale avait pour principal objectif la compréhension de l'intérieur de l'atome, c'est-à-dire du noyau, d'où l'emploi du qualificatif « nucléaire ».

Lorsque le Laboratoire vit officiellement le jour, en 1954, le Conseil provisoire fut dissous et la nouvelle organisation fut baptisée Organisation européenne pour la Recherche nucléaire. L'acronyme CERN fut toutefois conservé.

Aujourd'hui, notre compréhension de la matière va au-delà du noyau, et le principal domaine de recherche du CERN est la physique des particules, à savoir l'étude des composants fondamentaux de la matière ainsi que des forces auxquelles ils sont soumis. C'est pourquoi le CERN est souvent appelé « Laboratoire européen pour la physique des particules ». (CERN, 2013)

I.1.2 Les états membres (CERN, 2013)

Le CERN compte vingt États membres européens, mais de nombreux pays non européens participent également à ses activités, sous des formes diverses.

Actuellement, les États membres sont : l'Allemagne, l'Autriche, la Belgique, la Bulgarie, le Danemark, l'Espagne, la Finlande, la France, la Grèce, la Hongrie, l'Italie, la Norvège, les Pays-Bas, la Pologne, le Portugal, la République slovaque, la République tchèque, le Royaume-Uni, la Suède et la Suisse. La Roumanie a le statut de candidat à l'adhésion. Israël et la Serbie sont États membres associés en phase préalable à l'adhésion.

Les États membres ont des devoirs et des privilèges. Ils contribuent au budget et aux frais de fonctionnement des programmes du CERN ; ils sont représentés au Conseil, chargé des décisions importantes sur l'Organisation et ses activités.

Certains États (ou organisations internationales) qui, de façon permanente ou temporaire, ne peuvent pas devenir membres du CERN, ont le statut d'observateur. Ce statut permet

aux États non membres d'assister aux réunions du Conseil et de recevoir les documents du Conseil, sans prendre part aux décisions de l'Organisation.

Des scientifiques de quelque 608 instituts et universités du monde entier utilisent les installations du CERN.

Les physiciens et leurs organismes de financement, qu'ils soient d'États membres ou non, sont responsables du financement, de la réalisation et de l'exploitation des expériences auxquelles ils collaborent. Le CERN investit une part importante de son budget dans la construction de nouveaux accélérateurs, tels que le Grand collisionneur de hadrons LHC, et il ne peut contribuer que partiellement au coût des expériences.

Les États et les organisations ayant le statut d'observateur sont la Commission européenne, l'Inde, Israël, le Japon, la Fédération de Russie, la Turquie, l'UNESCO et les Etats-Unis d'Amérique.

Les États non-membres participant actuellement aux programmes du CERN sont l'Afrique du Sud, l'Algérie, l'Argentine, l'Arménie, l'Australie, l'Azerbaïdjan, le Belarus, le Brésil, le Canada, la Chine, Chypre, la Croatie, l'Estonie, la Géorgie, l'Inde, l'Iran, l'Irlande, l'Islande, le Maroc, le Mexique, le Pakistan, le Pérou, la Roumanie, la Serbie, la Slovénie, la Corée du Sud, Taiwan et l'Ukraine.

Le CERN a également des contacts scientifiques avec: l'Autorité palestinienne, la Chine (Taipei), Cuba, le Ghana, l'Irlande, la Lettonie, le Liban, Madagascar, la Malaisie, le Mozambique, les Philippines, le Qatar, le Rwanda, Singapour, le Sri Lanka, la Thaïlande, la Tunisie, l'Ouzbékistan et le Venezuela.

La moitié des physiciens du globe

Le CERN emploie 2400 personnes environ. Le personnel scientifique et technique du Laboratoire conçoit et construit les accélérateurs de particules et assure leur bon fonctionnement. Il contribue également à la préparation et à la mise en œuvre des expériences scientifiques complexes, ainsi qu'à l'analyse et à l'interprétation des résultats.

Environ 10,000 scientifiques visiteurs, soit la moitié des physiciens des particules du monde, viennent au CERN pour mener des recherches.

I.2 La structure du CERN (CERN, 2013)

La structure du CERN sous forme d'organigramme est présentée en annexe 1 de ce mémoire.

Le Conseil du CERN est l'autorité suprême et il est responsable de toutes les décisions importantes. Il détermine la politique de l'Organisation dans les domaines scientifique,

technique et administratif. Il approuve le programme d'activités, adopte les budgets et contrôle les dépenses.

Le Conseil est assisté dans sa tâche par le Comité des directives scientifiques et le Comité des finances.

Le Directeur général, nommé par le Conseil, gère le Laboratoire.

I.2.1 Le Conseil

Chaque État membre est représenté par deux délégués officiels au Conseil, dont l'un représente les autorités de son pays et l'autre les milieux scientifiques nationaux. Chaque État membre dispose d'une seule voix et la plupart des décisions se prennent à la majorité simple, même si, en pratique, le Conseil cherche à atteindre un consensus aussi large que possible.

I.2.2 Le Comité des directives scientifiques

Le Comité des directives scientifiques évalue l'intérêt scientifique des projets proposés par les physiciens et formule des recommandations sur le programme scientifique du CERN. Ses membres sont des scientifiques choisis par le Comité et nommés par le Conseil pour leurs compétences scientifiques, indépendamment de leur nationalité. Le Comité des directives scientifiques compte également des ressortissants d'États non-membres.

I.2.3 Le Comité des finances

Le Comité des finances réunit des représentants des autorités nationales et traite de toutes les questions relatives aux contributions financières des États membres, ainsi qu'au budget et aux dépenses de l'Organisation.

I.2.4 Le Directeur général

Nommé par le Conseil, habituellement pour cinq ans, le Directeur général administre le CERN. Le Directeur général est assisté par un Directoire composé de membres dont il propose la nomination au Conseil. Le Directeur général rend compte directement au Conseil auquel il peut proposer tout ajustement qu'il estime nécessaire.

I.2.5 Les Départements

Le CERN regroupe 8 départements qui sont divisés en groupes puis en sections. Les 8 départements du CERN sont les suivants :

PH - Physique - Chef : Philippe Bloch

IT - Technologies de l'information - Chef : Frederic Hemmer

BE - Faisceaux - Chef : Paul Collier

TE - Technologie - Chef : Frédérick Bordry

EN - Ingénierie - Chef : Roberto Saban

HR - Ressources humaines - Chef : Anne-Sylvie Catherin

FP - Finances, achats et transfert de connaissances - Chef : Thierry Lagrange

GS - Services généraux d'infrastructure - Chef : Thomas Pettersson

L'organigramme du département BE, dont je faisais parti lors de mon stage figure en annexe 2.

I.3 La physique (CERN, 2013)

Le programme de recherche du CERN aborde des thèmes variés allant des kaons aux rayons cosmiques et du Modèle standard à la supersymétrie.

La physique des particules – l'étude des constituants fondamentaux de la matière – constitue le principal axe des recherches menées au CERN. Toutefois, le programme de physique du Laboratoire est bien plus vaste, allant de la physique nucléaire à la physique des hautes énergies, et de l'étude de l'antimatière aux effets possibles des rayons cosmiques sur les nuages.

Depuis les années 1970, les physiciens des particules décrivent la structure fondamentale de la matière à l'aide d'une série d'équations complexes, constituant le Modèle standard. Ce modèle décrit le comportement des particules fondamentales - les constituants de base dont est fait toute chose visible dans l'Univers – ainsi que les quatre forces qui s'exercent entre elles. Les physiciens du CERN utilisent les accélérateurs et détecteurs de particules les plus puissants du monde pour mettre à l'épreuve les prédictions du Modèle standard. Avec le temps, ce modèle, qui a permis d'expliquer de nombreux résultats expérimentaux et a prédit avec précision toute une série de phénomènes, s'est imposé comme une théorie reposant sur de solides fondements expérimentaux.

Toutefois, le Modèle standard ne décrit que 4 % de l'Univers connu, et plusieurs questions restent sans réponse. Observerons-nous une unification des forces de la nature aux plus hautes énergies du Grand collisionneur de hadrons (LHC) ? Pourquoi la gravité est-elle si faible ? Pourquoi y a-t-il davantage de matière que d'antimatière dans l'Univers ? Doit-on s'attendre à découvrir une physique plus exotique à des énergies plus élevées ? Découvrirons-nous au LHC des éléments qui nous permettront de confirmer la théorie de la supersymétrie ? Ou de comprendre le boson de Higgs, qui confère la masse aux particules ?

Les physiciens du CERN cherchent les réponses à toutes ces questions, et bien plus encore :

À la recherche de l'antimatière

Le Big Bang devrait avoir créé matière et antimatière en quantités égales dans l'Univers. Alors pourquoi percevons-nous plus d'antimatière ?

Boson W : soleils et poussières d'étoiles

Le boson W est porteur de la force faible, l'une des quatre interactions fondamentales. Cette force peut changer la nature de particules de matière

Ions lourds et plasma de quarks et de gluons

Des collisions d'ions lourds pour libérer les quarks et recréer les conditions qui prévalaient juste après le Big Bang

La quête du boson de Higgs

Le 8 octobre 2013, le prix Nobel de physique a été attribué conjointement à François Englert et Peter Higgs « pour la découverte théorique du mécanisme contribuant à notre compréhension de l'origine de la masse des particules subatomiques et récemment confirmée par la découverte, par les expériences ATLAS et CMS auprès du LHC du CERN, de la particule fondamentale prédite par cette théorie ».

La supersymétrie

La supersymétrie prédit une particule partenaire pour chacune des particules du Modèle standard, pour expliquer d'où viens leur masse

La supraconductivité

En dessous d'une certaine température dite critique, un matériau n'oppose plus aucune résistance au passage d'un courant électrique

Le boson Z

Le boson Z est une particule élémentaire neutre, porteur de la force faible

Le Modèle standard

Le Modèle standard explique comment agissent les particules de matière, ainsi que leurs forces et particules porteuses

I.4 Les installations

Afin d'offrir les outils adéquats pour l'étude de la matière, le CERN développe et maintient deux types d'installations complémentaires : les accélérateurs qui permettent aux particules d'être accéléré avant leurs collisions et les expériences qui utilisent des détecteurs de collisions qui analysent le résultat des collisions.

I.4.1 Les accélérateurs (CERN, 2013)

Le complexe d'accélérateurs du CERN est une chaîne de machines qui accélèrent les particules à des énergies croissantes. Chaque machine augmente l'énergie d'un faisceau de particules avant de l'injecter dans la machine suivante. Dans le Grand collisionneur de hadrons (LHC) - le dernier maillon de la chaîne, les faisceaux de particules sont accélérés jusqu'à l'énergie record de 4 TeV par faisceau. La plupart des autres accélérateurs de la chaîne sont dotés de leur propre hall d'expérimentation, dans lequel les faisceaux sont utilisés pour des expériences réalisées à des énergies plus basses.

La source de protons est une simple bouteille d'hydrogène gazeux. Un champ électrique permet d'arracher à des atomes d'hydrogène leurs électrons et d'obtenir des protons. Le Linac2, premier maillon de la chaîne, accélère les protons jusqu'à une énergie de 50 MeV.



Figure 1 : Le Grand collisionneur de hadrons

Le faisceau est ensuite injecté dans le Synchrotron injecteur du PS (PS Booster - PSB), qui accélère les protons jusqu'à 1,4 GeV, puis dans le Synchrotron à protons (PS), qui porte le faisceau jusqu'à 25 GeV. Les protons sont ensuite envoyés au Supersynchrotron à protons (SPS), où ils sont accélérés à 450 GeV. Enfin, les protons sont envoyés dans les deux tubes de faisceau du LHC. Le faisceau circule dans le sens des aiguilles d'une montre dans le premier tube, et dans le sens inverse dans le deuxième. Il faut 4 minutes et 20 secondes pour remplir chacun des deux anneaux du LHC, et 20 minutes pour que les protons atteignent leur énergie maximale de 4 TeV. En conditions d'exploitation normales, les faisceaux circulent pendant plusieurs heures dans les tubes du LHC. Les deux faisceaux entrent en collision à l'intérieur de quatre détecteurs – ALICE, ATLAS, CMS et LHCb – ; au point de collision, l'énergie totale est de 8 TeV.

Le complexe d'accélérateurs, qui comprend également le Décélérateur d'antiprotons (AD) et le Séparateur d'isotopes en ligne ISOLDE, alimente aussi le projet Neutrinos du CERN vers le Gran Sasso (CNGS), la zone de test du Collisionneur linéaire compact (CLIC), ainsi que l'Installation de mesure du temps de vol des neutrons (nTOF).

Le LHC n'accélère pas que des protons. Les ions plomb du LHC sont issus d'une source de plomb vaporisé et passent par le Linac3 avant d'être collectés et accélérés dans l'Anneau d'ions de basse énergie (LEIR). Ensuite, le parcours au bout duquel les ions atteignent leur énergie maximale est le même que celui des protons.

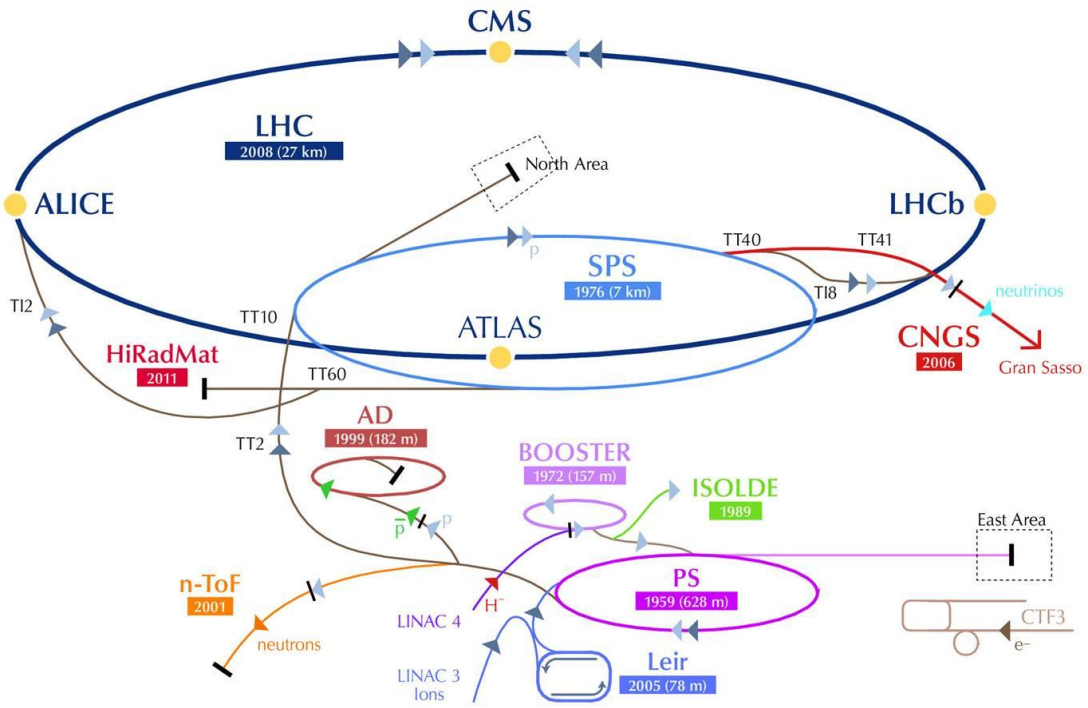


Figure 2 : schéma représentant les différents accélérateurs fonctionnant au CERN

Le Centre de contrôle

Le Centre de contrôle du CERN comprend les salles de contrôle des accélérateurs du Laboratoire, du système de distribution cryogénique et de l'infrastructure technique. Il compte 39 stations de travail pour quatre zones différentes : le LHC, le SPS, le PS et l'infrastructure technique.

I.4.2 Les expériences (CERN, 2013)

Parmi les expériences menées au Grand collisionneur de hadrons (LHC), sept utilisent des détecteurs pour analyser la myriade de particules produites lors des collisions dans l'accélérateur. Ces expériences sont conduites par des collaborations de chercheurs provenant d'instituts du monde entier, le CERN met a disposition l'infrastructure. Chacune est différente et se caractérise par ses détecteurs.

Les deux expériences les plus grandes, ATLAS et CMS, exploitent des détecteurs polyvalents pour explorer des domaines aussi vastes que possible. Le fait de disposer de deux détecteurs conçus indépendamment est essentiel pour recouper des informations en

cas de découverte. ALICE et LHCb utilisent des détecteurs spécialisés pour se concentrer sur des phénomènes particuliers. Ces quatre détecteurs sont logés dans d'énormes cavernes souterraines disposées le long de l'anneau du LHC.

Les plus petites expériences menées au LHC sont TOTEM et LHCf. Elles sont axées sur les particules dites « à très petits angles » – des protons ou des ions lourds qui se frôlent plutôt que d'entrer en collision frontale lorsque les faisceaux se croisent. TOTEM utilise des détecteurs placés de part et d'autre du point d'interaction de CMS, tandis que l'installation de LHCf est constituée de deux détecteurs disposés le long de la ligne de faisceau du LHC, à 140 mètres de chaque côté du point de collision d'ATLAS. MoEDAL, qui exploite des détecteurs déployés près de LHCb, a été conçu pour la recherche d'une particule hypothétique appelée monopôle magnétique.



Figure 3 : Le détecteur CMS

I.5 Le service informatique

Le département IT a pour mission de fournir la technologie de l'information nécessaire à l'accomplissement de la mission du laboratoire de manière efficiente et efficace à travers la consolidation de compétences de classe mondiale dans l'analyse technique, la conception, l'acquisition, la mise en œuvre, l'exploitation et le soutien de l'infrastructure et des services informatiques.

Le département IT du CERN est un environnement informatique très exigeant, qui maintien de vastes réseaux tant à l'échelle locale et mondiale, poussant les nouvelles technologies à leurs limites et en fournissant un terrain neutre pour la réalisation de la RD de pointe avec divers partenaires.

Parmi les principaux projets dont le département IT à la charge, on peut en citer deux principaux :

- Le Worldwide LHC Computing Grid (WLCG) : ce projet est une collaboration mondiale de plus de 170 centres de calcul dans près de 36 pays, reliant des infrastructures grid au niveau national et international.

  La mission du projet WLCG est de fournir des ressources informatiques globales pour stocker, distribuer et analyser les ~25 pétaoctets (25 millions de gigaoctets) de données par an générés par le Large Hadron Collider (LHC). (CERN, 2011)

- Le CERN openlab : il s'agit d'un partenariat unique public-privé entre le CERN et les principales entreprises de technologies de l'information et de la communication. Sa mission est d'accélérer le développement de solutions de pointe pour être utilisées par la communauté mondiale du LHC.

I.6 La section SU

Mon stage s'est déroulé dans la section SU du groupe ABP qui lui-même fait partie du département BE. En annexe 2 est présentée la structure du département BE, le groupe ABP en annexe 3 et enfin la section SU en annexe 4.

I.6.1 Présentation

La section SU (Large Scale Metrology) est responsable de la métrologie et de l'alignement des accélérateurs, des lignes de transfert de faisceaux associés et des détecteurs, pour l'ensemble du site du CERN.

Le mandat de la section SU comprend :

- Les aspects géodésiques des systèmes de référence et des algorithmes de traitement de données, ainsi que la création et l'entretien des réseaux géodésiques de référence,
- La définition des lignes de faisceaux dans le cadre du système de référence du CERN (CCS), ainsi que des principaux points d'implantation des bâtiments et tunnels abritant ces lignes de faisceaux,
- Les contrôles de qualité et les travaux métrologiques sur les composants du faisceau et des détecteurs lorsque les méthodes classiques de l'atelier ne peuvent pas être appliquées,
- L'alignement des éléments sur les lignes de faisceaux,
- L'infrastructure géométrique pour les détecteurs et leur métrologie pour leur montage sur les lignes de faisceaux,
- Les mesures sur les installations spécifiques et suivies de celle-ci lors des phases d'installation,
- La R & D liée à toutes ces tâches pour les machines actuelles et futures et des expériences et / ou mises à jour

Activités :

- Géodésie, analyse de données et stockage (base de données Survey)
- Métrologie des composants de faisceau (contrôles de qualité, contribution aux travaux de montage, fiducialisation)

- L'alignement des accélérateurs (complexe PS, nTOF, complexe SPS et lignes de faisceaux secondaire, CNGS, LEIR, LHC, Linac 4)
- Micro technologie et instrumentation (LHC low beta magnets, mise à jour du LHC, CLIC R & D, les installations d'étalonnage)
- Métrologie pour les détecteurs (CMS, Atlas, Alice, LHCb, TOTEM, CAST, les zones SPS, Compass, les zones PS et lignes de faisceaux secondaires)
- REX-ISOLDE, HIE-ISOLDE

I.6.2 Le service informatique de la section

La section SU regroupe trois sous-sections :

- AC pour 'Accelerator & Computing'
- EM pour 'Experiment Metrology'
- MTI pour 'Micro Technology and Instrumentation'

Mon stage s'est déroulé précisément dans la sous-section AC qui a pour mission de gérer la base de données de l'accélérateur et des logiciels d'analyse et de traitement de données.

Ma section a développé depuis un certain nombre d'années de nombreux logiciels dont la liste est présentée ci-dessous :

Base de données

Interfaces et bases de données de SU.

Base géodésique

Logiciels permettant les calibrations de l'instrumentation de SU.

Logiciels de calcul de formes

Logiciels pour trouver les paramètres de la forme, la surface, ou la ligne approchant au mieux un certain nombre de points ou d'objets. La majorité de ces programmes sont basés sur des algorithmes moindres carrés. Ils nous fournissent les meilleures estimations à partir d'informations redondantes ainsi que les statistiques pour l'analyse des données et des résultats.

Logiciels commerciaux

Logiciels du commerce à usage topographique, photogrammétrique ou géodésique au sein du groupe.

Librairies

Sous-routines, fonctions, procédures ou classes qui sont rassemblées dans une librairie de façon à être exploitées indépendamment par les logiciels.

Outils et utilitaires

Logiciels désignés à compléter la panoplie d'outils des programmes en fournissant soit des fonctionnalités nécessaires aux travaux quotidiens, soit une interface à un groupe de programmes qui doivent être utilisés conjointement pour accomplir une certaine tâche.

Logiciels de saisie et d'alignement

Logiciels développés uniquement pour des mesures géodésiques et des travaux d'alignement sur le terrain. Ces programmes sont conçus pour fonctionner sur PC portables ou carnets de terrain informatiques (type Husky) avec entrée des données par l'intermédiaire d'un clavier ou d'une connexion directe à l'instrument.

Logiciels de traitement et d'analyse

Logiciels développés pour traiter les données acquises sur le terrain et prodiguant les outils nécessaires à l'analyse des résultats. La majorité de ces programmes sont basés sur des algorithmes 'moindres carrés'. Ils nous fournissent les meilleures estimations à partir d'informations redondantes ainsi que les statistiques pour l'analyse des données et des résultats.

Logiciels de transformation

Logiciels développés afin de transformer des coordonnées de points ou des objets entre différents référentiels, et/ou déterminer les paramètres de la transformation entre deux référentiels cartésiens donnés, ou entre les référentiels géodésiques et les systèmes de projection utilisés au CERN. Les transformations sont réalisées par application de paramètres de transformation connus sur la liste initiale des points ou par déduction de ces paramètres à partir d'un certain nombre de points connus dans les deux référentiels puis par une application sur les points restants.

I.6.3 Le fonctionnement de la section SU

La section SU a pour mission d'assurer le parfait alignement des différents éléments composant les accélérateurs et les détecteurs selon les exigences définies par les physiciens pour un fonctionnement optimal des installations. Les coordonnées théoriques sont transmises par les physiciens à la section SU et sont intégrés à la base de données Survey. Grâce au système de référence du CERN appelé CCS (I.6.4.1), les géomètres de la section vont effectuer différentes mesures sur le terrain qui seront intégrés à la base de données. Les calculs effectués avec les mesures relevées sur le terrain permettrons de calculer les positions réelles des différents éléments composant les accélérateurs et les détecteurs et

ainsi, par moyen de leurs comparaisons aux positions théoriques, de réaliser les ajustements nécessaires sur ces installations.

Mon tuteur de stage, Mark Jones a la responsabilité de la partie informatique de la section SU – AC et aussi de la géodésie.

I.6.4 Le système métrologique du CERN

Afin de réaliser les différentes mesures de métrologie, le CERN dispose de son propre système de coordonnée.

I.6.4.1 Le système de coordonnée

Le système de coordonnée du CERN s'appelle CCS pour CERN Coordinate System. Le CCS est un système de coordonnées cartésiennes 3D étroitement lié à son système de référence. Il est utilisé pour définir la position relative des accélérateurs et des expériences du CERN. Le CCS a été créé pour le Synchrotron à protons (PS), le premier accélérateur du CERN, à la fin des années 1950. Le CCS est un système astronomique local modifié: le point principal est le pilier P0 au centre du PS; l'axe Z de ce système est, par définition, en coïncidence avec le vecteur de gravité à P0, et l'axe Y de le système à P0 a un azimut fixé par d'autres piliers géodésiques positionnés autour du PS.

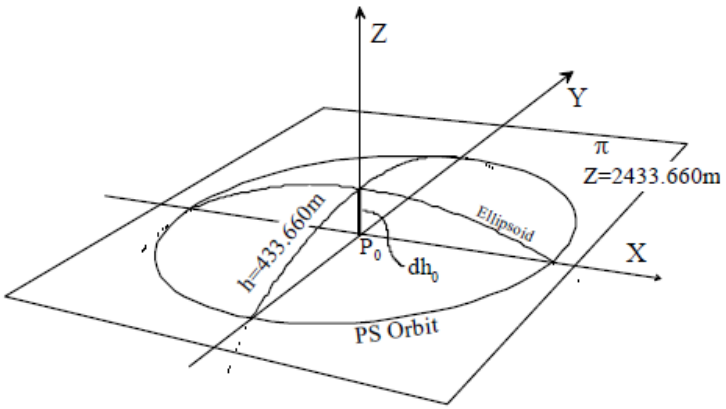


Figure 4 : Le système de coordonnées du CERN et de l'orbite PS

I.6.4.2 Le système géodésique de référence[1]

Dès les débuts du CERN, et des premiers accélérateurs, un réseau de piliers géodésiques a été établi sur le site et étendu, au fur et à mesure de l'extension du site.

[1] Beams Department The Newsletter does not necessarily reflect the views of the Beams Department The contributions solely reflect the views of their author(s) - Issue 3 NEWSLETTER December 2011

Les mesures classiques géodésiques ont été utilisées pour déterminer les coordonnées des piliers géodésiques impliquant: théodolites (pour mesurer directions / angles), lasers pour mesurer les distances, les niveaux optiques pour les hauteurs et les mesures des étoiles pour déterminer l'emplacement sur l'échelle mondiale. Même si le réseau couvrait une superficie d'environ 100 km2, il était le plus précis au monde (de l'ordre de 1 mm sur 10 km) et a été utilisée dans les années 1970 pour contrôler les mesures des premiers récepteurs GPS!

Pour le SPS et la machine LEP, les coordonnées des points de référence géodésique dans les tunnels des accélérateurs ont été déterminées au moyen d'un réseau de mesures de précision sur la surface à des points au-dessus des puits d'accès, suivies par leur transfert souterrain en utilisant des techniques optiques impliquant théodolites, plomb optique (essentiellement un télescope qui observe le long d'une ligne verticale), ou fils à plomb / pendules. Au moment des dernières mesures reliant la surface et les réseaux souterrains (durant les années 1980), la précision a été estimée à quelques millimètres.

Les points de référence dans les tunnels ont été établis à l'aide d'un réseau de mesures, incluant cette fois les mesures gyro-théodolites pour éviter les erreurs systématiques dues à la réfraction (où la ligne de visée entre deux points est courbée par les différences de température dans l'air). Enfin, ces points de référence dans les tunnels ont été utilisés pour aligner les éléments de la machine après qu'ils ont été installés, également avec une précision absolue estimée de quelques millimètres.

En utilisant les mesures GNSS[2], un lien entre le système de coordonnées CERN et un cadre de référence global a été créé au début du projet CNGS[3], puis contrôlé au cours des dernières années. La première campagne de mesures, à cinq piliers géodésiques, a été intégrée dans un calcul, par l'Office fédéral de topographie du réseau national suisse et leur position géodésique a été déterminée avec une précision globale de l'ordre de 7 mm. Cela a permis la localisation globale du site du CERN, qui doit être déterminée avec une précision d'environ 10 mm.

[2] Global Navigation Satellite System - un terme qui englobe GPS, Galileo, etc…

[3] Le projet Neutrinos du CERN vers le Gran Sasso (CNGS) vise à percer certains des mystères qui entourent les neutrinos, ces particules légères et neutres qui interagissent très peu avec la matière.

I.6.4.3 Autres systèmes de référence (Mark Jones, 2005)

Le Système géodésique de référence : CERN Geodetic Reference Frame est le référentiel qui existe implicitement dans la définition géodésique du CCS. Quand le SPS fut construit, la courbure de la Terre fut prise en compte par l'introduction d'une surface géodésique de référence de la forme d'une sphère. Pour le LEP, la sphère n'était plus une assez bonne approximation et un ellipsoïde bi-axial la remplaça comme surface géodésique de référence. Le CGRF a son origine au centre de cet ellipsoïde de référence, proche du centre de la Terre. L'altitude est mesurée a partir d'une surface de référence qui s'appel le géoïde, et non pas par rapport à l'ellipsoïde qui sert de surface de référence horizontale pour pouvoir déterminer les longitudes et latitudes.

I.7 Le stockage dans la base de données Survey

Toutes les données spatiales géométriques officielles des accélérateurs du CERN sont stockées en coordonnées CCS dans une base de données nommée Survey.

Une interface nommée Géode (développé avec ApEx[4]) donne accès aux logiciels de calculs tels que LGC[5] et Chaba[6] (écrits au sein de la section SU elles peuvent aussi fonctionner indépendamment) qui sont utilisés pour exploiter les mesures réalisées et déterminer les nouvelles coordonnées réelles spatiales des composants et leurs écarts par rapport aux positions théoriques. Les données sont enregistrées de manière à conserver les traces de toutes les mesures et de toutes les positions spatiales (théoriques et réelles) à un moment déterminées et sont ainsi exprimées dans un système de coordonnées stable et global pour l'ensemble des machines du CERN, ce qui permet de consulter à tout moment un historique des variations des points dans l'espace.

I.8 Les différents types de mesures

Lors de mon stage, j'ai principalement étudié les mesures de nivellement, d'offset (aussi appelé 'écartométrie') et de tilt. Ces trois types de mesure font partie intégrante du logiciel que j'ai étudié. Les membres de la section SU effectuent d'autre type de mesures qui

---

[4] Oracle Application Express, est un Environnement de développement intégré permettant de créer des applications de type web en développement rapide et dont le but est d'exploiter des bases de données Oracle.

[5] LGC (Logiciel Général de Compensation) est un outil pour le traitement des réseaux géodésiques du CERN.

[6] CHABA est un programme employé pour déterminer les paramètres de transformation qui lient deux jeux de coordonnées Cartésiennes pour un groupe de points communs, chacun déterminé dans un système de référence local différent.

seront, elles, stockées dans la base de données. Ces types de mesure, au nombre de 7, permettent d'assurer l'alignement de tous les composants, elles sont décrites dans cette partie.

I.8.1 Le nivellement

Le nivellement consiste à mesurer l'altitude des points. Cette mesure de nivellement est généralement réalisée à l'aide d'un niveau (optique, automatique, etc…) et consiste à mesurer le dénivelé (hauteur) entre deux points, par rapport à un plan local. En parcourant les points les uns par rapport aux autres, on peut ainsi déterminer la différence de hauteur entre deux points éloignés. Si l'un des deux points a une altitude connue, on peut ainsi déterminer l'altitude du second.

Une opération de nivellement se réalise au préalable en préparant 'une station', c'est-à-dire l'endroit où va être installé le niveau et par rapport à cet emplacement, quels points vont être mesurés. Après avoir mesuré tous les points, on mesure de nouveau tous les points dans le sens inverse afin de vérifier s'il n'y a pas d'erreur de lecture sur les mesures 'aller'[7].



Figure 5 : Mesure de nivellement sur un dipôle du LHC

Si les mesures 'aller' et 'retour'[8] sont correctes, les mesures enregistrées seront la moyenne entre l'aller et le retour. S'il y a une trop grande différence entre les mesures aller et retour, une reprise de la station est nécessaire (c.-à-d. une troisième mesure) afin de déterminer quel point pose problème et à quelle étape de la mesure (aller ou retour).

[7] Sens initial de prise de mesures d'une station
[8] Mesures réalisées dans le sens contraire du sens initial

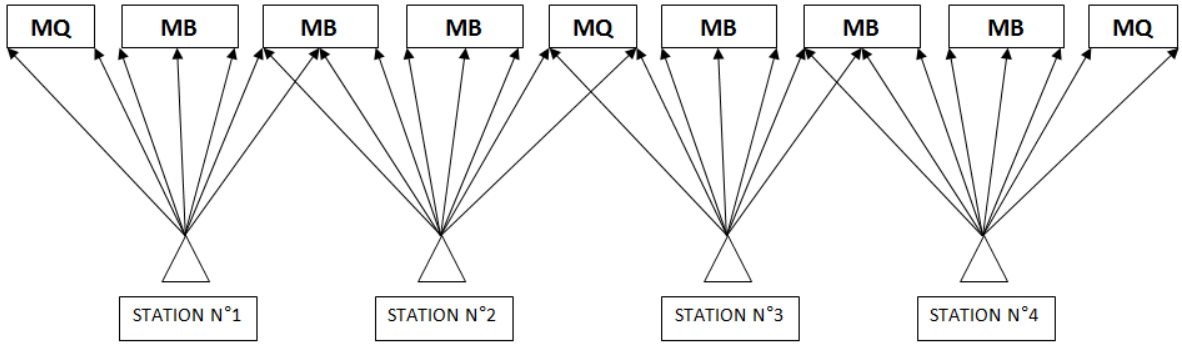


Figure 6 : Stations de mesures

I.8.2 L'offset

L'offset a pour but de mesurer l'écart d'un point par rapport à un point vertical. Cette droite est matérialisée par un fil tendu entre deux points (la tension dépendant de la précision recherchée et du type de fil). La mesure d'offset se fait alors avec un écartomètre, d'où le nom de la technique : écartométrie. Les écartomètres peuvent être de différente nature selon la précision recherchée et les impératifs budgétaires : règles "manuelles" (glissement d'un chariot capteur sur une règle encodée, style pieds à coulisse), règles automatiques (comme celles développées au CERN dans les années 70, utilisant des diodes électroluminescentes pour détecter un fil et se placer à sa verticale), capteurs capacitif comme les WPS (Wire Positioning System) installés notamment dans le LHC, au niveau des aimants low-bêtas proches des expériences, etc.

Le but de l'offset est de réaliser un alignement horizontal entre les aimants afin que les particules circulent dans les meilleures conditions possibles dans l'accélérateur.

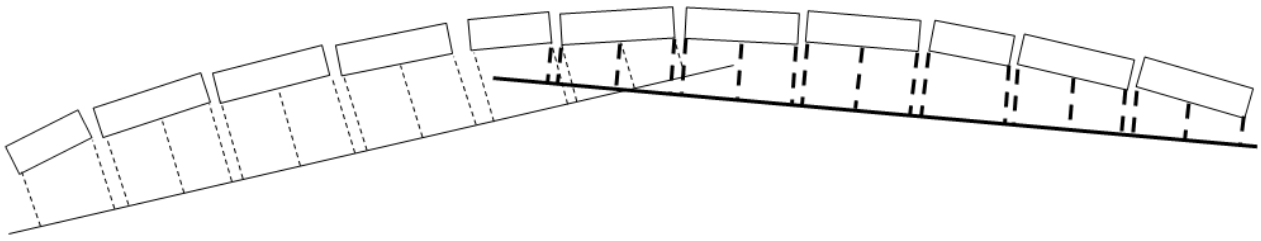


Figure 7 : Fils de mesures

I.8.3 Le tilt

Les mesures de tilt consistent à mesurer l'angle (du plan principal) d'un composant par rapport à l'horizon local, dans le sens transversal au faisceau.

I.8.4 Autres mesures

La mesure d'angle vertical

La mesure d'angles verticaux consiste par exemple à mesurer l'angle entre le plan horizontal passant par un point stationné par l'instrument et la droite visant un autre point. Les mesures d'angles verticaux peuvent servir, combinées avec la mesure de distance, à déterminer indirectement le dénivelé entre deux points.

La mesure d'angle vertical utilisée pour les angles horizontaux

La mesure d'angles horizontaux utilise la même procédure que précédemment, mais avec un plan vertical. Le plan vertical utilisé comme référence est défini par l'instrument (le zéro de de son cercle horizontal).

La mesure de distance

Une mesure de distance consiste à mesurer la distance entre deux points, dans le cas le plus basique, avec un ruban-mètre, dans d'autres cas, selon la précision recherchée, avec d'autres types d'instrument (électro-distance mètres, mékomètres, lasers, capteurs capacitifs, etc.). Selon les instruments, la mesure de distance peut se faire sans contact (mesures capacitives ou laser) ou sur des réflecteurs & prismes (mesures EDM ou laser).

Les mesures gyroscopiques

Les mesures gyroscopiques consistent à mesurer l'azimut (par rapport au nord géographique) d'une direction entre le point stationné par le gyro-théodolite (il s'agit d'un gyroscope monté sur un théodolite mesurant les angles horizontaux) et un point visé.

Ces types de mesures étant peu précises, elles sont utilisées non pas pour affiner les orientations mais pour les contrôler.

II Description du projet

Le but de cette partie est de présenter le projet de développement sur lequel j'ai travaillé en présentant les aspects techniques ainsi que les difficultés prévisibles.

II.1 Cahier des charges

A ce jour, un logiciel développé en interne par la section SU assiste les géomètres lors de leurs prises de mesures des éléments des accélérateurs. Les mesures de nivellement, écartométrie et tilt, essentiellement sur les dipôles et quadripôles, sont réalisées via les appareils adéquats puis enregistrées grâce au logiciel PFB. PFB a été développé en langage Visual Basic et fonctionne sur la plate-forme Windows CE, ceci pour un usage mobile via un PDA supportant cet OS.

Windows CE est aujourd'hui obsolète et a été supplanté par iOS (Apple) et Android (Google). Les applications iOS sont programmées en C, C++ et Objective-C tandis que celles d'Android sont programmées en C, C++ et Java (Briot, 2012) .

Outre iOS et Android, 4 autres OS animent actuellement des appareils mobiles : Blackberry OS (Blackberry), Symbian (Nokia), Windows Phone (Microsoft) et Bada (Samsung).

Symbian et Blackberry OS sont en perte de vitesse et les applications qui s'exécutent sur chacun d'eux sont programmées respectivement en C++ et Java. Windows Phone et Bada sont des challengers dans l'univers des OS mobile et les applications qui s'exécutent sur l'un et l'autre sont programmées en C# et VB.Net pour Windows Phone et C++ pour Bada.

Tous ces OS sont mutuellement incompatibles, d'où la nécessité de coder une application distincte pour chaque plate-forme mobile.

Un des buts du nouveau développement du logiciel PFB est d'assurer sa portabilité sur n'importe quel appareil mobile, quel que soit l'OS. Une des solutions est de passer par le développement d'une 'WebApp', un navigateur Web étant juste nécessaire au fonctionnement de l'application.

Néanmoins il y a des limitations de l'approche Web App. Les navigateurs des téléphones sont très hétérogènes et ont chacun une implémentation différente des standards du Web :

- Différences de capacité entre le navigateur d'un Feature Phone et celui d'un Smartphone
- Différence de comportement entre les navigateurs de différents Smartphones pour la même fonctionnalité

Dans le cas du choix d'un développement par WebApp, il conviendra de tester le comportement du logiciel sur un maximum de plateformes différentes afin de vérifier la bonne exécution du logiciel sur chacune d'elles.

Une autre solution serait de passer par un compilateur 'cross-plateforme' qui permet, à partir d'un seul code source, de compiler une application destinée à plusieurs plateformes cibles, on est ici dans le cas d'une application native.

En outre, quelle que soit la solution retenue, le logiciel devra assurer les fonctionnalités suivantes :

- Un enregistrement des données de mesures en local (pas de perte de données en cas d'absence de réseau Internet)
- La récupération de données pour la prise de mesure (donnée relative aux points à mesurer, appareils de mesure utilisée…) localement sur l'appareil mobile.

Enfin, dans le cas où le choix se porterait sur une solution 'propriétaire', celle-ci devra être viable à long terme (choisir une solution déjà adopté par le CERN).

II.2 Spécificité du développement

II.2.1 Documentation existante

Le principal point, qui a été une vraie difficulté au début de mon projet, a été le manque de documentation sur le logiciel PFB actuel. En effet celle-ci est très limitée, se résumant à la documentation utilisateur. Par ailleurs, lors de mes premières lectures du code source, je me suis aperçu que celui-ci était très peu commenté, quelques phrases étant écrites ça et là mais rien sur la philosophie de fonctionnement du logiciel en général. Ainsi le temps passé sur la phase d'analyse du logiciel a été importante, car s'articulant autour de la lecture du code source (~12000 lignes de code).

II.2.2 Séparation entre les données et la vue (GUI)

Le second point à prendre en compte : le logiciel développé devra intégrer une séparation entre l'interface graphique et les données manipulées, dans le but de pouvoir réutiliser aisément le code écrit pour l'adapter à d'autres terminaux ou logiciels.

III Analyse de l'existant

La première phase de mon travail a consisté à analyser le code du logiciel PFB existant pour en retranscrire son fonctionnement et surtout d'établir quelles données sont manipulées et comment. En effet, PFB repose essentiellement sur l'usage de fichiers 'data' (fichiers textes) ayant pour extension '.dat' en entrée et après réalisation des mesures, à la création de fichiers 'data', toujours avec l'extension '.dat' en sortie. Ces fichiers de sorties seront utilisées pour l'intégration des mesures dans la base de donnée Survey.

III.1 Le logiciel PFB

Le logiciel PFB pour Portable (Survey) Field Book fonctionne sur PocketPc et assiste les géomètres dans leurs prises de mesures. Dans cette partie est détaillée l'interface graphique du logiciel et pour déterminer son fonctionnement interne, des diagrammes de séquences seront détaillés.

III.2 L'interface graphique

Dans un premier temps, lors du démarrage de l'application, un menu apparait nous demandant quel type de mesures nous voulons réaliser. Le deuxième menu, qui apparait quel que soit le type de mesure choisi est le réglage des références. Il s'agit ici de spécifier le nom de l'équipe réalisant les mesures, l'instrument utilisé, la mire utilisée (pour le nivellement uniquement), le fichier théorique, de séquence (pour le nivellement et l'écartométrie) et le numéro d'opération (deux numéros pour le tilt, l'initial et le 'après réglage').

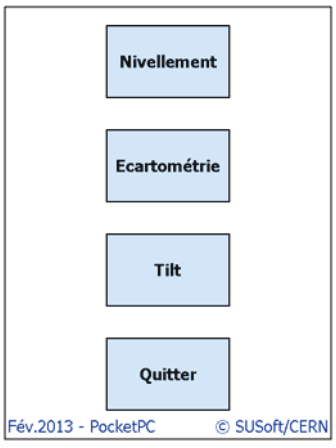


Figure 8 : Menu principal de PFB

Ce réglage des références est primordial et va conditionner les mesures qui vont être réalisées. En effet, le fichier théorique et le fichier de séquence provient de la base de données Survey et ils déterminent pour le fichier théorique, le positionnement idéal des différents éléments tandis que le fichier de séquence détermine lui une séquence de mesure logique afin que les équipes effectuent les mesures de façon efficaces. Ce réglage de référence a aussi pour but de préciser le matériel utilisé. Selon le type d'appareil renseigné, le logiciel fait apparaitre ou non les menus destinés à la communication afin que les mesures soient prises de façon automatique (communication Bluetooth).

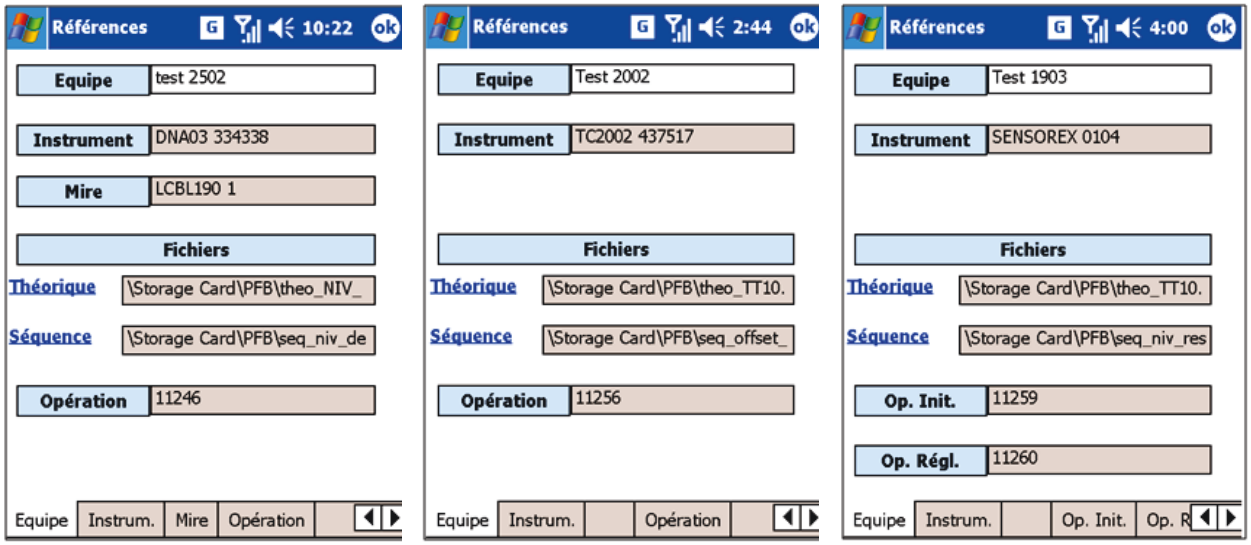


Figure 9 : Réglage des références - Module de nivellement, d'écartométrie et tilt

Voici ci-dessous un exemple de fichier théorique où sont enregistrées les informations qui seront ensuite utilisé par le logiciel comme le nom complet du point (Zone, Classe, Numéro et Type), les coordonnées selon le CERN Coordinate System, etc

Zone/Accl	Classe Point	No Point	Mire Point	Dist. Cumul.	X	Y	Z	H	Dist. Inter.	tilt	pente	gisement	Code	Type	Date	n° série élément	Rallonge tracés
LHC	MQML	8R6	E	16964.28113	2449.96505	10364.98703	2365.328	370.70671	1.898	0.00086	-0.013363	125.3875	1	A	08-Feb-12	HCLQ.NCA001-CR000662	
LHC	MQML	8R6	T	16967.98673	2453.58393	10364.03329	2365.27771	370.65545	99.999	0.00086	-0.013363	125.3875	1	O	08-Feb-12	HCLQ.NCA001-CR000662	
LHC	MQML	8R6	S	16967.98687	2453.37816	10363.54454	2365.27919	370.65628	4.8	0.00086	-0.013363	125.3875	1	O	08-Feb-12	HCLQ.NCA001-CR000662	
LHC	MB	A9R6	E	16972.63779	2457.66142	10361.73573	2365.21352	370.58857	2.407	0.00274	-0.013363	125.5499	1	A	08-Feb-12	HCLBARE000-IN001316	
LHC	MB	A9R6	M	16978.0369	2462.63164	10359.62815	2365.13948	370.51216	99.999	0.00274	-0.013363	125.5499	1	O	08-Feb-12	HCLBARE000-IN001316	
LHC	MB	A9R6	S	16983.43688	2467.60115	10357.51656	2365.06917	370.43948	14.3	0.00274	-0.013363	125.5499	1	O	08-Feb-12	HCLBARE000-IN001316	
LHC	MB	A9R6	T	16983.43737	2467.80935	10358.00599	2365.06706	370.43802	99.999	0.00274	-0.013363	125.5499	1	O	08-Feb-12	HCLBARE000-IN001316	
LHC	MB	B9R6	E	16988.29704	2472.06772	10355.60655	2365.00273	370.3709	1.361	-0.00038	-0.01334	125.8746	1	A	08-Feb-12	HCLBBRS000-IN003309	
LHC	MB	B9R6	M	16993.69941	2477.02867	10353.46899	2364.92867	370.29445	99.999	-0.00038	-0.01334	125.8746	1	O	08-Feb-12	HCLBBRS000-IN003309	

Tableau 1 : Données figurant dans un fichier théorique

Pour le fichier de séquence, les informations sont moins nombreuses. Les champs de données correspondent à des séquences de mesure types, tel que la configuration des lieux et la dispositions des éléments le permet. Il est utilisé principalement dans le module Nivellement et Ecartométrie pour définir une station de mesure dans le module nivellement et un fil de mesure pour le module écartométrie.

Station	Zone/Accl	Classe Point	No Point
25	LHC	MQMC	9R6E
25	LHC	MQMC	9R6S
25	LHC	MB	A10R6E
25	LHC	MB	A10R6M
25	LHC	MB	A10R6S
25	LHC	MB	B10R6E
25	LHC	GGPSO	10R6
26	LHC	GGPSO	10R6
26	LHC	MB	A10R6M
26	LHC	MB	A10R6S
26	LHC	MB	B10R6E
26	LHC	MB	B10R6M
26	LHC	MB	B10R6S
26	LHC	MQML	10R6E
26	LHC	MQML	10R6S

Tableau 2 : Données figurant dans un fichier séquence de nivellement

Pour le module de nivellement, il contient, dans l'ordre, les informations suivantes :

- Le numéro de la station
- La zone ou se trouve l'élément, ici le LHC.

LHC	MQM	6L8SLHC	MB	B9L8ELHC	MQM	6L8E	-.05183
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MQM	B7L8S	-.15400
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MQM	B7L8E	-.18638
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	A8L8S	-.21762
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	A8L8E	-.26420
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	B8L8S	-.27308
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	B8L8E	-.26461
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MQML	8L8S	-.25744
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MQML	8L8E	-.24517
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	A9L8S	-.22412
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	A9L8E	-.16007
LHC	MQM	6L8SLHC	MB	B9L8ELHC	MB	B9L8S	-.12019
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	A8L8E	-.14968
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B8L8S	-.20496
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B8L8E	-.29960
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MQML	8L8S	-.33684
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MQML	8L8E	-.35994
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	A9L8S	-.37946
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	A9L8E	-.41853
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B9L8S	-.42503
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B9L8E	-.40794
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MQMC	9L8S	-.39761
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MQMC	9L8E	-.38055
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	A10L8S	-.35148
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	A10L8E	-.28109
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B10L8S	-.23537
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MB	B10L8E	-.10761
LHC	MB	A8L8SLHC	MQML	10L8ELHC	MQML	10L8S	-.05093

Tableau 3 : Données figurant dans un fichier séquence d'écartométrie

- La classe du point (ou type d'élément), ici MQMC ou MQML pour Magnet Quadripole (MB et ML sont des variantes de ce type d'aimant) ou MB pour Magnet Bending (aimant destiné à courber le faisceau de particule). Les points GGPSO sont des points de référence géodésique qui se trouve au sol.
- Le numéro du point, selon le système d'identification établi par le CERN pour tous les éléments du LHC. Le numéro intègre aussi la mire du point.

Pour le module d'écartométrie, il contient, dans l'ordre, les informations suivante :

- La zone ou se trouve l'élément, ici le LHC.
- La classe du point (ou type d'élément), généralement sur les cloisons ou au sol.
- Le numéro du point

Sur chaque ligne se trouvent les noms de trois points. Le premier correspond à l'ancrage du fil, le suivant au deuxième ancrage et le troisième au point à mesurer avec son écart théorique.

Les réglages des instruments, mires et opérations font appel à des structures en arbre. Pour les instruments, selon le modèle sélectionné, il apparait le choix de l'utiliser en mode connecté ou non (si l'appareil en question supporte le mode connecté).

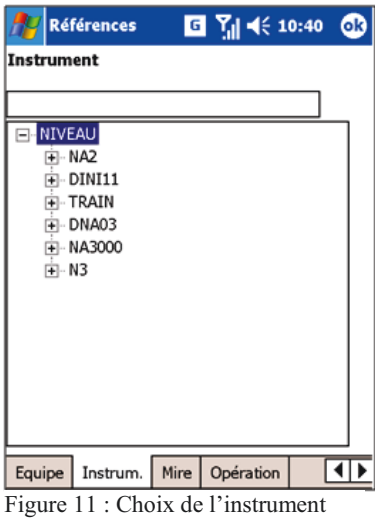


Figure 11 : Choix de l'instrument

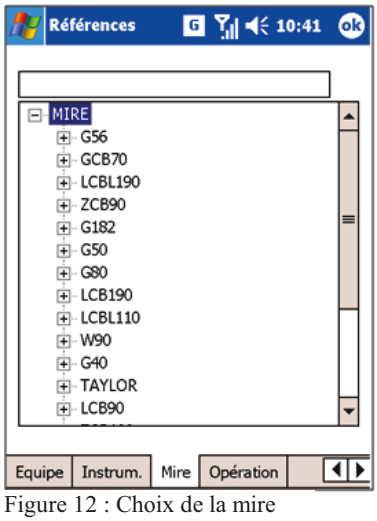


Figure 12 : Choix de la mire

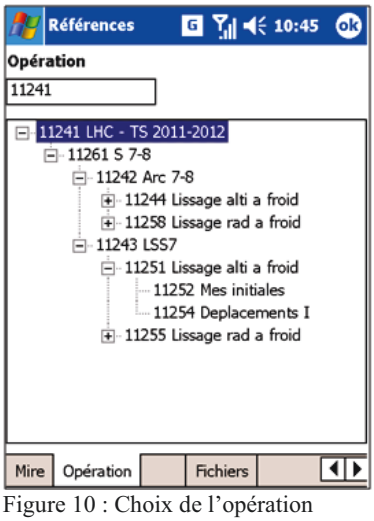


Figure 10 : Choix de l'opération

Lorsque les références sont toutes renseignées, l'utilisateur appuis sur OK et la fenêtre de réglage disparait pour laisser place à la fenêtre destinée à la prise de mesure.

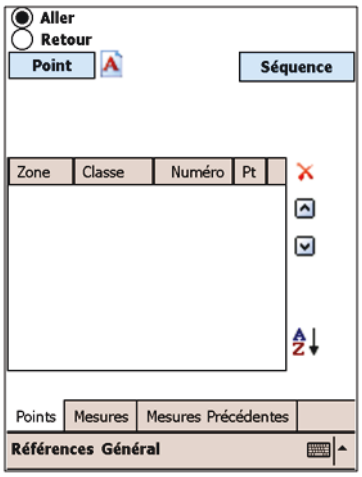


Figure 13 : Module de nivellement - Sélection des points

La première fenêtre qui apparait dès lors est celle qui permet de définir les points qui devront être mesuré. Il y a dès lors plusieurs manières de sélectionner les points : la première, par le bouton point qui affichera une nouvelle fenêtre présentant les points présents dans le fichier théorique sélectionné précédemment ; la deuxième, par l'ajout de points manuellements en cliquant sur l'icône 'A' ; la troisième, par l'appui sur le bouton 'Séquence' qui fait apparaitre les listes des séquences de mesures. Dès que l'on a sélectionné une séquence, le tableau affichant les séquences disparait et apparait de nouveau la liste des points correspondant à la séquence choisie. Quelle que soit la méthode choisie, il est toujours possible de supprimer un point (icône 'X'), de changer leurs ordres (icones ^ et v) et de trier la liste (icone az).

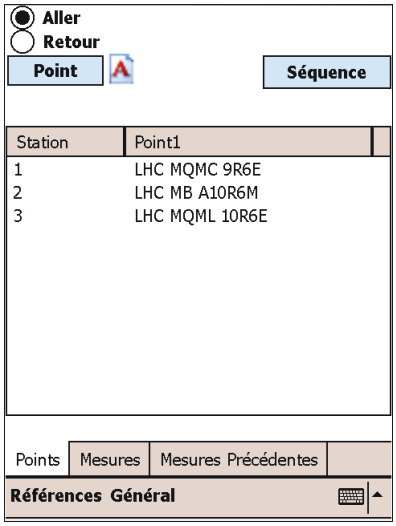


Figure 15 : Choix d'une séquence (station)

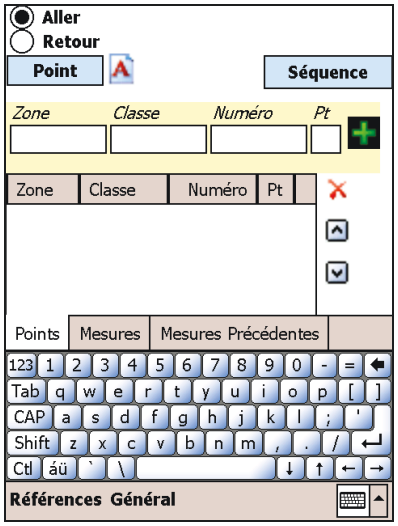


Figure 16 : Ajout manuel d'un point

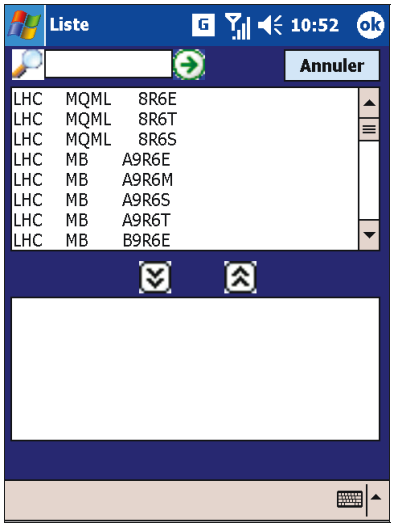


Figure 14 : Choix des points

L'utilisateur appuie ensuite sur l'onglet 'Mesure' afin de réaliser les différentes mesures. Pour le module 'Tilt' il existe deux onglets de mesures, le premier nommé 'Initial' car corespondant à des mesures réalisées avant d'effectuer des changements de position de l'élément mesuré et le second 'Réglage' qui correspond à des mesures réalisées après déplacement de l'élément mesuré.

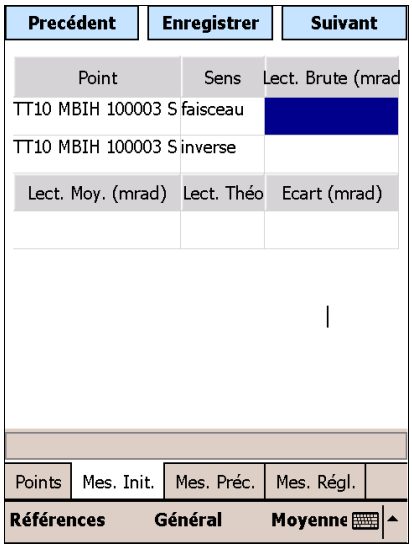


Figure 17 : Mesures de tilt

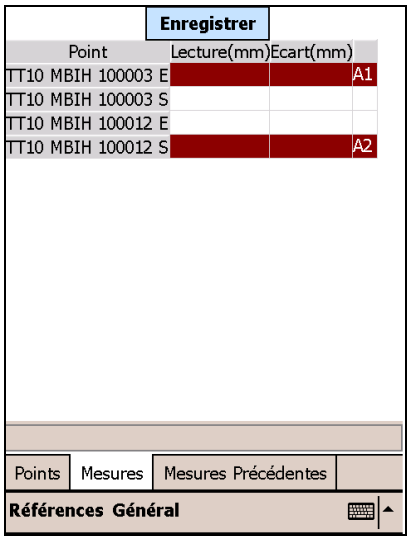


Figure 19 : Mesures d'écartométrie

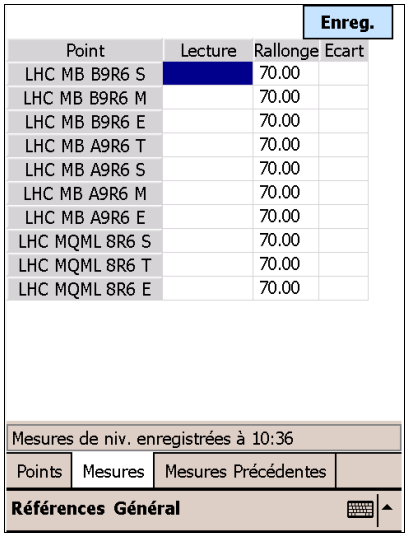


Figure 18 : Mesures de nivellement

Si un appareil connecté a été selectionné, il n'est pas possible d'éditer les cellules. Des boutons de contrôle apparaissent pour réaliser les mesures par communications RS232 (le PocketPc et le dispositif de mesure sont préalablement relié via Bluetooth). Ces boutons (purge, init et mesurer) sont destinés à gérer la communication avec l'appareil de mesure.

Enfin, lorsque l'on appuie sur le bouton 'enregistrer', les mesures sont enregistrées dans un fichier texte (avec l'extention .dat) pour pouvoir être récupérées par la suite et insérées dans la base de donnée Survey. Pour le module nivellement, un autre fichier est créé en cas de prise de mesure 'Aller' (fichier .txt). Il est destiné à enregistrer les stations 'Aller' afin que les mesures 'Retour' puissent être réalisées.

Les données figurant dans les fichiers de sortie correspondent aux mesures réalisées sur le terrain et disposent des éléménts suivants :

<u>Pour le module de nivellement :</u>

- Le numéro d'opération, le nom de l'équipe qui a réalisé la mesure, la date de la réalisation de la mesure
- Le point mesuré
- L'appareil de mesure et la mire utilisés
- La valeur de la mesure réalisée
- La dimension de la rallonge utilisée
- Le type de station (Aller, Retour ou Reprise)

- Le calcul de l'emq[9]

```
000;RE;23-Sep-2013;TESTDATA  ; 11632;
001;PH;DINI11    ;106746     ;;;;;;;;;;;;;;
002;LE;GCB70     ;2          ;TT10   ;MAL   ;102928;S;    1452; 0.07000;;;;;;A;        ;;;;;16:10:02;
003;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;E;    1362; 0.00000;;;;;;A;        ;;;;;16:10:02;
004;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;S;    1425; 0.00000;;;;;;A;        ;;;;;16:10:02;
005;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;E;    1362; 0.00000;;;;;;A;        ;;;;;16:10:02;
006;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;S;    1456; 0.00000;;;;;;A;        ;;;;;16:10:02;
007;PH;DINI11    ;106746     ;;;;;;;;;;;;;;
008;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;S;    1462; 0.00000;;;;;;R;        ;;;;;16:10:54;emq= 44.2
009;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;E;    1362; 0.00000;;;;;;R;        ;;;;;16:10:54;emq= 44.2
010;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;S;    1420; 0.00000;;;;;;R;        ;;;;;16:10:54;emq= 44.2
011;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;E;    1463; 0.00000;;;;;;R;        ;;;;;16:10:54;emq= 44.2
012;LE;GCB70     ;2          ;TT10   ;MAL   ;102928;S;    1463; 0.07000;;;;;;R;        ;;;;;16:10:54;emq= 44.2
013;PH;DINI11    ;106746     ;;;;;;;;;;;;;;
014;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;S;    1452; 0.00000;;;;;;P;        ;;;;;16:11:49;emq= 1.7
015;LE;GCB70     ;2          ;TT10   ;BPCL  ;102963;E;    1362; 0.00000;;;;;;P;        ;;;;;16:11:49;emq= 1.7
016;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;S;    1423; 0.00000;;;;;;P;        ;;;;;16:11:49;emq= 1.7
017;LE;GCB70     ;2          ;TT10   ;BSG   ;102937;E;    1360; 0.00000;;;;;;P;        ;;;;;16:11:49;emq= 1.7
018;LE;GCB70     ;2          ;TT10   ;MAL   ;102928;S;    1452; 0.07000;;;;;;P;        ;;;;;16:11:49;emq= 1.7
```

Tableau 4 : Module de nivellement - Exemple de fichier de sortie .dat

```
 000  SPS   QF    32010E    25511   0.07000
 000  SPS   QF    32010S    25489   0.07000
 000  SPS   QD    32110E    25500   0.07000
 000  SPS   QD    32110S    25478   0.07000
 001  SPS   QD    32110E    31266   0.07000
 001  SPS   QD    32110S    31236   0.07000
 001  SPS   QF    32210E    31226   0.07000
 001  SPS   QF    32210S    31226   0.07000
 002  SPS   QF    32210E    32012   0.07000
 002  SPS   QF    32210S    32006   0.07000
 002  SPS   QD    32310E    31952   0.07000
 002  SPS   QD    32310S    31960   0.07000
```

Tableau 5 : Module de nivellement - Exemple de fichier 'Retour.txt'

<u>Pour le module d'écartométrie :</u>

- Le numéro d'opération, le nom de l'équipe qui a réalisé la mesure, la date de la réalisation de la mesure
- Les points d'ancrages utilisés (A1 et A2)
- Le point mesuré
- L'appareil de mesure utilisé
- La valeur de la mesure réalisée
- Le calcul de l'écart par rapport aux données théoriques

[9] Erreur quadratique moyenne

```
000;RE;19-Mar-2013;Test 2002 ; 11256;
001;A1;         ;            ;TT10   ;MBIH  ;100003;E;        ;;;;;;;;
002;A2;         ;            ;TT10   ;MAL   ;100103;S;        ;;;;;;;;
003;OF;TC2002   ;437517      ;TT10   ;MBIH  ;100003;S; 0.11000;        ; 0.00000;;;;;;;;;;14:21:01;
004;OF;TC2002   ;437517      ;TT10   ;MBIH  ;100012;E; 0.14300;        ; 0.00000;;;;;;;;;;14:21:01;
005;OF;TC2002   ;437517      ;TT10   ;MBIH  ;100012;S; 0.16400;        ; 0.00000;;;;;;;;;;14:21:01;
006;OF;TC2002   ;437517      ;TT10   ;QIID  ;100100;E; 0.12100;        ; 0.00000;;;;;;;;;;14:21:01;
007;OF;TC2002   ;437517      ;TT10   ;QIID  ;100100;S; 0.11400;        ; 0.00000;;;;;;;;;;14:21:01;
008;OF;TC2002   ;437517      ;TT10   ;MAL   ;100103;E; 0.11100;        ; 0.00000;;;;;;;;;;14:21:01;
009;EC;TT10   ;MBIH  ;100003;S;   0.00001;          ;          ;          ;
010;EC;TT10   ;MBIH  ;100012;E;  -0.00026;          ;          ;          ;
011;EC;TT10   ;MBIH  ;100012;S;   0.00064;          ;          ;          ;
012;EC;TT10   ;QIID  ;100100;E;   0.00025;          ;          ;          ;
013;EC;TT10   ;QIID  ;100100;S;  -0.00004;          ;          ;          ;
014;EC;TT10   ;MAL   ;100103;E;   0.00069;          ;          ;          ;
```

Tableau 6 : Module d'écartométrie - Exemple de fichier de sortie .dat

Pour le module de tilt :

- Le numéro d'opération, le nom de l'équipe qui a réalisé la mesure, la date de la réalisation de la mesure
- Le point mesuré
- L'appareil de mesure utilisé
- La valeur des mesures réalisées (sens faisceau et inverse)
- Le calcul de l'écart par rapport aux données théoriques

```
000;RE;19-Mar-2013;Test 1903 ; 11259;
001;AT;SENSOREX ;0104  ;TT10   ;MBIH  ;100003;S; 0.0000000;;;;;;;;;;;;15:17:58;sens faisceau= 0.00mrad    sens inverse= 0.00mrad
002;EC;TT10   ;MBIH  ;100003;S;         ;         ;         ;   0.00002;
```

Tableau 7 : Module de tilt - Exemple de fichier de sortie .dat

III.3 Le Data Workflow

Le Data Workflow des prises de mesure est assez simple et permet au géomètre de bien préparer les mesures à réaliser. Les fichiers théoriques et de séquences sont générés par le logiciel Géode qui accède à la base de données Survey. Ensuite le géomètre insère les fichiers dans son terminal mobile afin de réaliser les mesures. Une fois les mesures réalisées, il ne reste plus au géomètre qu'à insérer dans la base de données les fichiers correspondant aux prises de mesures.

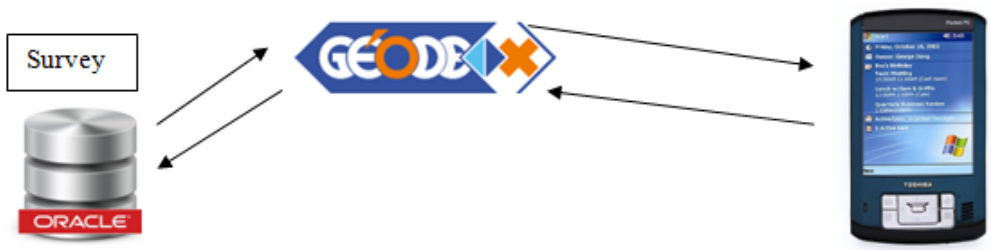


Figure 20 : Prise de mesures - Data Workflow

IV Choix de la plateforme de développement

Un des points clés de la réécriture du logiciel PFB se situe au niveau de la portabilité. En effet, il m'a été demandé à ce que le nouveau logiciel puisse fonctionner sur différentes plateformes mobiles. Cette partie traite de l'étude de la problématique et du choix retenu.

IV.1 Les différentes solutions existantes

Il y a principalement deux types de solutions pour développer un logiciel multiplateforme, soit développer une 'WebApp', c'est-à-dire une application qui s'ouvre par le biais d'un navigateur, soit développer une application native (qui utilise les fonctions natives de l'appareil) en utilisant un Framework de développement multiplateforme qui compilera un logiciel écrit dans un certain langage (comme C++ ou JavaScript/CSS/HTML5) pour une plateforme spécifique comme Android ou iOS.

IV.1.1 L'approche WebApp

Ces applications sont nettement moins performantes que les applications natives, car il faut payer le coût du navigateur web, mais elles ont l'avantage de pouvoir fonctionner facilement sur un nombre bien plus important de plateformes. Il existe de nombreux Framework pour le développement d'une WebApp (CRAZY WS, 2012).

IV.1.1.1 jQuery Mobile

Touch-Optimized Web Framework pour les Smartphones et tablettes. C'est un système unifié compatible avec la majorité des appareils mobiles.
Il est basé sur jQuery et jQuery UI. Et avec son write less, do more, plus besoin d'avoir à développer votre site mobile pour tel ou tel périphérique! jQuery Mobile fournit une bibliothèque de composants suffisante mais un résultat esthétique plus éloigné de l'environnement natif. Il est intéressants par exemple dans le cas où l'application n'a pas de besoins d'aspect natif (ideo MOBILIS, 2012).

IV.1.1.2 Sencha Touch

Sencha Touch est le tout premier Framework pour le développement d'applications web sur mobile. Permet de développer pour presque n'importe quel type de périphérique comme Android ou iPhone.

Sencha Touch est la version mobile du bien connu ExtJs. Cette solution est la plus indiquée si l'on souhaite réaliser une application HTML/Javascript esthétiquement identique à une application mobile native. Il est particulièrement exhaustif et ses thèmes iOS et Android sont soignés. Il est construit sur une architecture objet Javascript entièrement créée par l'équipe de Sencha et permet d'utiliser le design pattern « Model View Controller » pour une meilleure maintenabilité et une meilleure évolutivité. Un peu de temps est requis pour le prendre en main car si une utilisation basique est plutôt bien documentée, la consultation des sources est nécessaire pour traiter les situations plus compliquées. Mais le forum est assez réactif bien que la priorité de l'équipe Sencha aille au support payant (ideo MOBILIS, 2012).

IV.1.1.3 Titanium

Titanium est une API et une plateforme JavaScript éditées par « Appcelerator ». Il permet de créer des applications natives et offre une API supportant une grande quantité de ressources matérielles. Appcelerator fournit également un IDE basé sur Eclipse, « Titanium studio », qui permet de compiler et tester son code à la volée. Le codage est rapide et il suffit de peu de code pour développer une application.
Avec plus de 300 APIs et une équipe de développeurs florissante, vous pourrez créer des applications mobiles extensibles.

IV.1.1.4 PhoneGap

Phonegap est une solution hybride. C'est une bibliothèque permettant d'accéder depuis le JavaScript d'une page web à de nombreuses ressources matérielles de l'appareil. Le système de callback sur lequel il fonctionne pose des problèmes lors de son intégration avec des bibliothèques JavaScript tierces. Phonegap ne gère pas l'interface utilisateur qui est entièrement laissée à la charge de l'utilisateur et du navigateur embarqué. La présentation se fait donc à l'aide de CSS, de bibliothèques JavaScript tierces, ou des langages spécialisés tels que Mobl (ideo MOBILIS, 2011).

IV.1.1.5 Rhodes

"Rhodes" est le framework que propose Rhomobile pour le développement d'applications mobiles. Le développement se fait dans Eclipse à l'aide du plugin "Rho Studio" et du debugger "Rho Debugger" à installer sur l'appareil de test. Une application "Rhodes" est une application hybride de type "Ruby On Rails" pouvant accéder à certains widgets et aux

ressources matérielles des appareils. On peut le combiner avec un framework HTML tel que Sencha pour compléter les éléments graphiques qu'il propose (ideo MOBILIS, 2011).

IV.1.2 La solution ‘application native’

Les solutions natives ont pour la plupart accès à un niveau de performance permettant de créer des applications riches incluant des animations ou des ergonomies originales.

IV.1.2.1 Qt

Qt est un Framework de développement complet avec des outils conçus pour simplifier la création d'applications et d'interfaces utilisateur pour les ordinateurs de bureau, embarqués et mobiles.

- Qt Framework - API intuitive pour C++ et CSS/JavaScript- comme la programmation avec Qt Quick pour la création rapide de l'interface utilisateur
- IDE Qt Creator – puissant environnement multiplateforme de développement intégré, y compris les outils de conception d'interface utilisateur et de débogage
- Outils et toolchains - support pour l’internationalisation, boite à outils embarqués et plus encore.

Avec Qt, on peut réutiliser le code pour cibler efficacement plusieurs plates-formes avec une base de code. La conception modulaire des bibliothèques de classes C+ et des outils de développement permet aux développeurs de créer facilement des applications pour une plate-forme et de créer et exécuter le déploiement sur une autre plateforme. (Qt Project, 2013) Le support d’Android et de iOS est prévu pour automne 2013 avec la version Qt 5.2.

IV.1.2.2 Java (Virtual Machine sur OS Mobile)

Les agendas électroniques (BlackBerry, Palm), tablettes (iPad, Android), smartphones (iPhone, Android), consoles de jeu (Nintendo Wii), lecteur MP3/MP4 (iPod) et autres dispositifs mobiles personnels ne prennent pas en charge le plug-in Java. (ORACLE, s.d.)
Java pour les dispositifs mobiles s'appuie sur la plate-forme Java Platform, Micro Edition (Java ME).
Néanmoins, Java ME n'est pas utilisée sur les plus récentes plates-formes mobiles comme iPhone, Windows Phone, BlackBerry 10 et Android. Il continue cependant d'être très populaire dans les appareils de moins de 200 $ comme les Nokia Series 40. Il est également utilisé sur le nouveau système d'exploitation Bada ainsi que sur Symbian OS avec un logiciel natif (WIKIPEDIA, 2013).

IV.1.2.3 MoSync SDK (MoSync AB, 2013)

MoSync est un Framework cross-plateforme de développement mobile et permet à tout développeur C et ou C++, Lua et Javascript de créer des applications mobiles pour les plateformes Android, Symbian, J2ME, iOS, BlackBerry, Windows mobile et Windows Phone 7 avec le même code source.

De nos jours, à part quelques Frameworks qui font du natif, les solutions cross plateformes dans le domaine du développement mobile orientent vers l'utilisation du HTML5/JavaScript/CSS3 qui ne fournit qu'un accès limité aux fonctionnalités natives de l'appareil. Avec MoSync, on a plusieurs choix :

- utiliser le C et le C++ uniquement pour créer des applications avec interfaces natives ou personnalisées (au choix)
- utiliser HTML5 et un Framework JavaScript quelconque pour créer des applications mobiles hybrides
- utiliser HTML5, C++ plus la technologie JavaScript Wormhole de MoSync pour franchir les limites d'accès du HTML5 à certaines fonctionnalités des appareils en communiquant entre le JavaScript et le C++ (avec Wormhole), ce dernier vous permettant d'avoir accès à toutes les fonctionnalités natives du téléphone
- utiliser le HTML 5 ou le JavaScript uniquement (comme le cas du Framework) pour créer des interfaces natives avec un accès complet aux fonctionnalités natives du téléphone.

MoSync n'est pas une solution de développement mobile cross plateforme comme les autres. Elle est particulière sur plusieurs points. La plupart des solutions cross plateformes créent des applications qui sont interprétées par des machines virtuelles, ce qui réduit d'ailleurs les performances. MoSync ne supporte pas que cette seule technologie pour rendre ses applications multiplateformes. En fonction de chaque type de plateforme, MoSync a une approche différente (et la meilleure possible) pour s'exécuter au mieux sur cette plateforme (Gbofu, 2012).

IV.1.2.4 Haxe (Haxe, 2013)

Haxe est un langage de programmation open source.

Tandis que les langages de programmation actuel s’appuie sur leur propre plateforme (Java avec la JVM, C# avec .Net et Action Script avec Flash Player), Haxe est un langage multiplateforme.

La syntaxe de Haxe est similaire à Java, Action Script et C++.

Ce logiciel libre permet de générer du code source dans un langage cible plutôt que du code binaire.

La communauté Haxe maintient un nombre impressionnant de bibliothèques. Certaines permettent même de convertir le modèle objet d'une plateforme vers une autre au moment de la compilation. Ce qui a pour but de compiler un même code Haxe vers plusieurs langages cible (WIKIPEDIA, 2013).

De plus, Haxe permet de générer du code C++ à partir de son code source, ce qui est très utile pour créer des applications natives, pour l'iPhone par exemple.

IV.2 Choix de la solution

A ce jour il n'y a pas de solution unique pour développer une application multiplateforme mobile mais des solutions existent et le meilleur choix sera dépendant des besoins exprimés.

Dans le cas d'une application native développée à l'aide d'un outil cross-plateforme, la performance, l'utilisation des ressources matérielles et le respect du design sera au rendez-vous. Néanmoins, si l'on ne considère qu'un seul développement, l'on reste dépendant de l'outil cross-plateforme utilisé. Ensuite, la maintenance est plus compliquée car une modification du code originel impose de réaliser un redéploiement sur tous les appareils mobiles utilisés.

Les solutions Web hybrides permettent de créer des applications qui se déploieront facilement sur une grande quantité de plateformes et elles utilisent des techniques de développement très standard (CSS, Javascript et HTML5). De plus, la maintenance et l'évolutivité est très simplifiée car une modification sur le code se réplique immédiatement sur tous les appareils mobiles lors de la prochaine reconnexion. Autre caractéristique intéressante, le serveur Web récupère en temps réel les informations saisies sur l'appareil client (pas de besoin de consulter les données sur chaque appareil).

Néanmoins, ce type d'applications est nettement moins performant que les applications natives car il faut payer le coût du navigateur Web et elles ne peuvent pas présenter d'aspect parfaitement fidèle à l'aspect natif. Mais si l'on décide de mettre en place un aspect spécifique, on peut présenter de très belles interfaces. Enfin, des solutions existent permettant d'avoir accès aux ressources matérielles de l'appareil, il convient simplement de bien veiller à ce que le Framework choisi prenne bien en charge celles dont on aura besoin (www.markus-falk.com, s.d.) (WIKIPEDIA, 2013).

Dernier point qui est primordial dans le cadre de l'usage d'une application Web Mobile, le stockage de donnée sur le poste client. Ceci a été rendu possible grâce à l'API HTML5 'Web Storage' qui permet une utilisation hors ligne de l'application et donc de stocker les données saisies en local et de les synchroniser avec le serveur une fois connecté (Alexandre, 2012).

Le choix final s'est finalement porté sur la solution Qt pour les raisons suivantes :

- Le logiciel est destiné à être utilisé le plus souvent dans des zones où il n'y a pas de connexion interne et l'approche WebApp ne permet pas de garantir une persistance des données au delà de 5 Mo utilisé dans le 'Web Storage', ce qui est très limité pour le chargement complet du programme et des données associées.
- En mettant de côté la solution WebApp, le choix se porte sur Qt car ce Framework est très populaire auprès des développeurs et les cycles de sortie des nouvelles versions sont régulières et apportent à chacune son lot de nouveautés.
- Ce Framework de développement multiplateforme est largement éprouvé sur l'univers Desktop (Windows, OSX et Linux) et a été choisi par Canonical pour le développement d'Unity 8 (Unity Next), la version desktop d'Ubuntu Touch (Ubuntu pour Smartphones et tablettes).
- C'est le Framework utilisé dans la section Survey pour le développement des applications et dans un souci d'uniformité de plateforme de développement, c'est la solution privilégié.
- Le seul point négatif est le support non complet des univers mobiles (iOS et Android) au début de mon projet. Néanmoins, j'ai pu commencer à développer sur Windows et par la suite compilé le code source pour le système Android sans problème.

IV.3 Le Framework Qt (Digia Oyj, 2013)

Qt est un Framework [10]de développement multiplateforme avec des outils conçus pour simplifier la création d'applications natives et des interfaces utilisateur pour ordinateur de bureau, embarqué et des plates-formes mobiles.

[10] Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

Qt est leader dans l'industrie du développement d'applications et permet aux développeurs de cibler des univers très différents (mobile, desktop…) avec une seule base de code.

Il offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt permet de concevoir des interfaces graphiques par son architecture en utilisant les mécanismes des signaux et slots par exemple.

La question principale à se poser lors du démarrage d'un développement d'une application avec Qt est l'interface graphique à utiliser. En effet, Qt dispose de deux possibilités, la classe QWidget qui est la classe de base de tous les objets d'interface utilisateur et QML. Le widget est l'atome de l'interface utilisateur : il reçoit des événements de souris, de clavier et d'autres depuis le système de fenêtrage, et peint une représentation de lui-même à l'écran. Cette classe est essentiellement utilisée dans les environnements de bureau et est par exemple sur laquelle repose l'environnement graphique KDE, l'un des plus utilisés dans le monde Linux.

QML pour Qt Meta Language est un langage déclaratif basé sur JavaScript pour concevoir des applications d'interface centrées sur les utilisateurs. Il fait partie de Qt Quick, le kit de création d'interface utilisateur développée par Nokia dans le Framework Qt. QML est principalement utilisé pour les applications mobiles où la saisie tactile, les animations fluides (60 FPS) et l'expérience utilisateur est cruciale.

Au premier abord, on peut penser que la solution à privilégier pour la réécriture du logiciel PFB est d'utiliser QML car il s'agit d'une solution adaptée à l'univers mobile. Cependant, les QWidget ont été conçus pour un type d'interface utilisateur différent de QML. Les QWidgets présentent un meilleur choix si l'UI à développer est constituée d'un petit nombre d'éléments complexes et statiques et QML est un meilleur choix si l'UI est constituée d'un grand nombre d'éléments simples et dynamiques.

En d'autres termes, étant donné que l'on aura à gérer plusieurs tableaux mais d'une certaine complexité, la solution QWidget est à privilégier. D'autant plus que la classe QWidget dispose déjà de nombreux widgets permettant une gestion simple de tableaux, listes, affichage des structures en arbres, etc… Le but de l'application à réécrire n'est pas d'avoir des transitions spectaculaires entre les fenêtres mais bien d'avoir une présentation adéquate des données manipulées.

IV.3.1 Les signaux et les slots

Le principal élément à prendre en compte pour la réécriture du logiciel sous Qt est que la GUI (Graphic User Interface) est totalement à repenser. Les autres classes (ou objets) seront similaires entre le logiciel VB.Net et Qt, car ce sont deux langages objets et ces classes ont été créées spécifiquement pour répondre aux besoins 'métier'. Par exemple la classe 'Point' dans VB.Net est destinée à encapsuler toutes les données concernant les points à mesurer (leurs noms, les mesures…).

La GUI sous Qt fonctionne avec deux éléments fondamentaux, les signaux et les slots. Le signal est un message envoyé par un widget (fenêtre ou élément composant une fenêtre) lorsqu'un événement se produit. Le slot est une fonction qui est appelée lorsqu'un événement ce produit et donc appelle une méthode d'une classe. Pour résumer on peut dire qu'un signal appelle un slot.

Exemple de code intégrant un slot et un signal :

```
connect(ui->butSupprimer, SIGNAL(clicked()), this, SLOT(actionSupprimer()));
```

Dans l'exemple ci-dessus, le mot clé 'connect' indique que l'on veut connecter un signal à un slot. Le premier élément à l'intérieur des parenthèses correspond au widget qui envoi le signal, ici le bouton 'butSupprimer' du GUI. Le troisième élément indique dans quelle classe le slot va être appelé. Enfin le dernier élément correspond à la méthode appelée par le signal.

Enfin Qt permet de connecter un signal à plusieurs slots, donc un click sur un bouton peut appeler plusieurs méthodes. Il est par ailleurs possible de connecter un signal à un autre signal.

IV.3.2 L'architecture MVC

L'architecture MVC, pour modèle-vue-contrôleur de Qt, sépare les éléments du programme en 3 parties :

- Le modèle qui contient les données.
- La vue qui gère l'affichage par le biais des widgets, donc de ce que contient le modèle.
- Le contrôleur qui est la partie algorithmique, qui réalise des calculs sur les données par exemples.

En pratique il y a que deux niveaux sur Qt, le modèle et la vue. Le contrôleur est en fait intégré à la vue.

Par exemple, si l'on veut créer un tableau puis insérer des données à l'intérieur, on procèdera comme suit :

- On crée un modèle grâce à la classe QStandardItemModel qui est la classe la plus adaptée pour gérer les données d'un tableau.
- On crée la vue grâce à la table QTableView.
- On associe la vue et le modèle.

Qt permet d'associer un même modèle à plusieurs vue et inversement d'avoir plusieurs modèles appliqués à une seule vue.

Cette architecture MVC de Qt sera utilisée en permanence lors de la réécriture afin de répondre au cahier des charges de la réécriture du logiciel (II.1).

IV.3.3 Qt Créator et Qt Designer

La réécriture a été entièrement réalisée à l'aide de Qt Créator, qui est IDE[11] idéal pour programmer des projets Qt. Qt Créator s'accompagne de Qt Designer qui permet de réaliser la GUI sous forme graphique de façon rapide et efficace, Qt Designer regroupe les zones suivantes :

- Une barre d'outils qui permet d'édité les widgets, les signaux/slots…
- Une zone centrale où l'on dessine les fenêtres.
- Une zone regroupant tous les widgets existants (bouton, tableau, …) afin de les intégrer dans la fenêtre en cours.
- Une zone 'property editor' qui permet de modifier les propriétés du widget sélectionné.
- Une zone 'object inspector' qui affiche la liste des widgets placés sur la fenêtre sous forme d'arbre.
- Un éditeur de signaux et de slots et d'actions.

Un exemple de fenêtre d'édition Qt Créator figure en annexe 5.

[11] Integrated Development Environment (Environnement de développement intégré)

V Conception et implémentation

V.1 Initialisation sous forme de génie logiciel

Lorsque l'étude du logiciel original de PFB était proche d'être terminée, il était préférable de démarrer la réécriture sur des bases de travail provenant du génie logiciel. Ainsi, avant de démarrer directement la réécriture, j'ai estimé l'effort à produire pour la réécriture et, à partir de là, de découper le projet en étape.

V.2 Estimation de l'effort à produire pour le développement

V.2.1 Méthode des points de fonction

La méthode des points de fonction permet de mesurer la taille d'un logiciel en dénombrant les fonctionnalités offertes aux utilisateurs. Cette méthode est plus fiable que la méthode COCOMO mais sa précision est obtenue pour des projets regroupant plusieurs milliers de points de fonction.

V.2.1.1 Dénombrements des différents composants

Le dénombrement des composants figure en annexe 6, il regroupe les éléments suivant :

- Les groupes de données internes (GDI), c'est-à-dire les données gérées par l'application.
- Les groupes de données externes (GDE), c'est-à-dire les données non gérées par l'application mais uniquement consultées.
- Les entrées de données (ENT) relatives aux créations, modifications, validations, suppressions de données.
- Les sorties de données (SOR) relatives aux restitutions des données transformées.
- Les interrogations (INT) relatives aux consultations de données (listes, détails…).

Les composants ont été dénombrés par l'analyse du point de vue utilisateur de l'application PFB actuelle.

V.2.1.2 Calcul des points de fonction brut

En annexe 7, chaque composant est pondéré par rapport à sa complexité et son type. Ainsi, l'on dénombre au total 179 points de fonction bruts. Le nombre donné dans la colonne 'Total par complexité' est donnée par la méthode des points de fonction.

V.2.1.3 Calcul des points de fonction ajusté

Les points de fonction dénombrés sont dès lors ajustés par rapport à un facteur d'ajustement lié aux exigences du développement, comme par exemple des critères de performances, la facilité d'exploitation, etc..

- L'addition de tous ces critères nous permet de calculer le facteur d'ajustement par la formule FA = 0,65 + (0,01 x somme des facteurs).
  Ce qui donne : FA = 0,65 + (0,01 x 18) = 0,83
- Les points de fonction ajusté sont dès lors calculé par la formule : PFA = points de fonction bruts x FA
  Ce qui donne : PFA = 179 x 0,83 = 149

Calcul du facteur d'ajustement					
F1 télécommunications	(1)	2	3	4	5
~~F2 traitement distribué~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
~~F3 performance requise~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
~~F4 configuration saturée~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
~~F5 taux de transaction~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
F6 saisie interactive	1	(2)	3	4	5
F7 efficacité utilisateur	1	2	(3)	4	5
~~F8 m-a-j interactive~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
~~F9 traitement complexe~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
F10 réutilisabilité	1	2	3	(4)	5
F11 facilité d'installation	1	2	3	(4)	5
F12 facilité d'exploitation	1	2	3	(4)	5
~~F13 plusieurs sites~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
~~F14 configurable~~	~~1~~	~~2~~	~~3~~	~~4~~	~~5~~
Total colonne					
Total	18				

Figure 21 : Calcul du facteur d'ajustement

V.2.1.4 Calcul de la charge de travail

La charge de travail est estimée par le biais du tableau ci-contre. Étant donné que la taille du projet est estimée comme petite, la charge est égale à 149 x 2 = 298 journée.homme. Cette charge est une nouvelle fois à pondérer, car s'agissant d'une réécriture de logiciel, la charge sera inférieure à celle prévisible d'un projet démarré à partir d'une feuille blanche. Ainsi, il est raisonnablement envisageable que le temps de développement sera inférieur à cette valeur de 298 journée.homme, mais il reste à savoir dans quelle proportion.

Calcul de la charge de travail	
Taille du projet	Charge en journée.homme par PFA
Petit	2
Moyen	3
Grand	4

Figure 22 : Calcul de la charge de travail

V.2.2 Méthode COCCOMO

Cette méthode est relativement simple mais est aussi moins précise que la méthode des points de fonction. Cependant, étant donné que le logiciel a déjà été développé, nous pouvons reprendre le nombre de lignes de code du logiciel existant, qui est composé d'environ 12000 lignes de code.

Ainsi, par COCCOMO, nous calculons d'abord l'effort à fournir, qui, pour un projet simple est calculé par la formule : Effort = 2,4 x KLS[12] exposant 1,05 ce qui donne un effort égal à 2,4 x 12 exposant 1,05 = 32,61.

Le temps de développement est donnée par la formule : TDev = 2,5 x Effort exposant 0,38 ce qui donne un temps de développement égal à 2,5 x 32, 61 exposant 0,38 = 9,40 mois.

De même que pour la méthode des points de fonction, nous pouvons penser que le temps développement sera plus court, mais l'on doit mettre en place un planning pour prendre en compte cet état de fait.

V.3 Mise en place d'un planning de développement

A la date de l'étude du temps de développement, il me restait 119 jours de travail, ce qui représentait moins de la moitié du temps nécessaire suite aux évaluations réalisées précédemment. Néanmoins, nous nous attendions que le temps de développement soit plus court que celui calculé, du fait que nous nous basions sur un logiciel existant, mais nous ne savions pas dans quelle proportion. Nous avons donc décidé de réaliser un planning de développement 'par module' sur le temps restant pour réécrire le logiciel et comprenant à chaque fois une phase de conception générale, détaillée, de codage et tests unitaires et, pour finir, d'intégration et de tests fonctionnels. Pour chaque module, le temps de développement de chaque phase est donnée par la méthode COCCOMO (WIKIPEDIA, 2013), soit 16% du temps en phase de conception générale, 26% en phase de conception détaillée, 42% en phase de programmations et tests unitaires et enfin 16% en phase d'intégration et tests fonctionnels.

Le planning élaboré figure en annexe 8 sous forme de diagramme de Gantt. Le développement par module permettra d'ajuster le planning prévisionnel au planning réellement réalisé et ainsi de modifier le planning prévisionnel des modules suivants en conséquence. Par cette méthode, au fur et à mesure de l'avancement du projet, nous étions à même d'estimer plus finement le temps de développement du module suivant et de situer dans quelle proportion le temps de développement serait plus court que celui calculé précédemment. En annexe 9, 10 et 11 sont présentées des comparaisons des plannings initialement estimés pour chaque module avec ce qui a réellement été réalisé.

[12] pour Kilo Ligne Source

L'on peut voir que le temps de développement lié à l'écriture du code et aux tests pour le premier module (Tilt) a été plus important que prévu. Ensuite, grâce à l'expérience acquise sur le module de Tilt (programmation avec Qt, réutilisation de certaines fonctions communes), le module de nivellement a été écrit dans les temps annoncés, sauf pour la partie communication (Bluetooth). Pour finir, grâce aussi au fait qu'il y a des nombreuses similitudes entre le module de nivellement et d'écartométrie, le module d'écartométrie à été réalisé dans des temps records, ce qui m'a permis d'être en phase avec mon planning dans l'ensemble.

V.3.1 Utilisation de JIRA

Afin de centraliser les informations liées au développement des logiciels dans la section SU, il a été décidé que le système permettant la gestion de projet soit un outil déjà largement utilisé au CERN : JIRA.

JIRA gère le cycle complet du développement d'application et j'ai donc utilisé cet outil pour partager mon planning de développement, que je mettais à jour régulièrement, les documents réalisés, etc…

V.4 Politique de développement par module

Afin de réaliser un développement le plus efficace possible, la réécriture du logiciel a été divisée par module et dans la méthode cela se rapproche de ce qui se fait en XP (Extreme Programming) dont les principaux points sont les suivants :

- Le développement est réalisé en collaboration étroite avec le client. Lors de mon stage, je développais dans les mêmes bureaux que les clients (les géomètres), la collaboration étroite était de facto établie.
- Une première version du logiciel apparait très tôt et les nouvelles versions s'enchainent à un rythme soutenu. De par le mode de développement par module, les utilisateurs ont eu rapidement une version à tester.
- A chaque fonction écrite, j'ai réalisé des tests pour vérifier qu'elle avait le comportement attendu.
- Le planning de développement théorique est réévalué à la fin de la réalisation de chaque tâche, de manière à ce que les temps de développement pour les tâches suivantes soient le plus juste possible.

V.5 La conception générale et détaillée

Suite au temps passé sur la lecture du code orignal de PFB codé en VB.NET, j'ai formalisé mes notes sous forme de diagramme UML, principalement des diagrammes d'activités

pour indiquer l'ordre de déroulement des méthodes appelées par le logiciel. Le comportement du logiciel devant être identique entre l'ancienne version et la nouvelle, les diagrammes utilisés pour la nouvelle version sont identiques.

De même les diagrammes de séquences n'ont été écrits qu'une seule fois et découlent du logiciel original. Les langages VB.NET et C++ étant tout deux des langages objets, les différences entre les diagrammes de séquences sont minimes car ils permettent de représenter des collaborations entre objets d'un point de vue temporel.

Le nombre de diagrammes de séquences présentés n'est pas exhaustif, j'ai volontairement choisi de présenter les principaux diagrammes et des diagrammes différents pour chaque module car les diagrammes pour la même activité sont relativement similaires entre les modules.

V.5.1 Les diagrammes d'activité – Module de tilt

Les diagrammes d'activité pour le module de tilt se trouvent en annexe 12, le surlignage correspond aux interactions de l'utilisateur avec l'interface graphique, les autres interactions correspondant aux interactions entre les méthodes.

Outre la validation des références qui va déclencher l'appel de diverses méthodes, l'utilisateur peut par la suite sélectionner le point à mesurer, soit à partir d'une liste prédéfinie (créée à partir du fichier théorique), soit le rajouter manuellement.

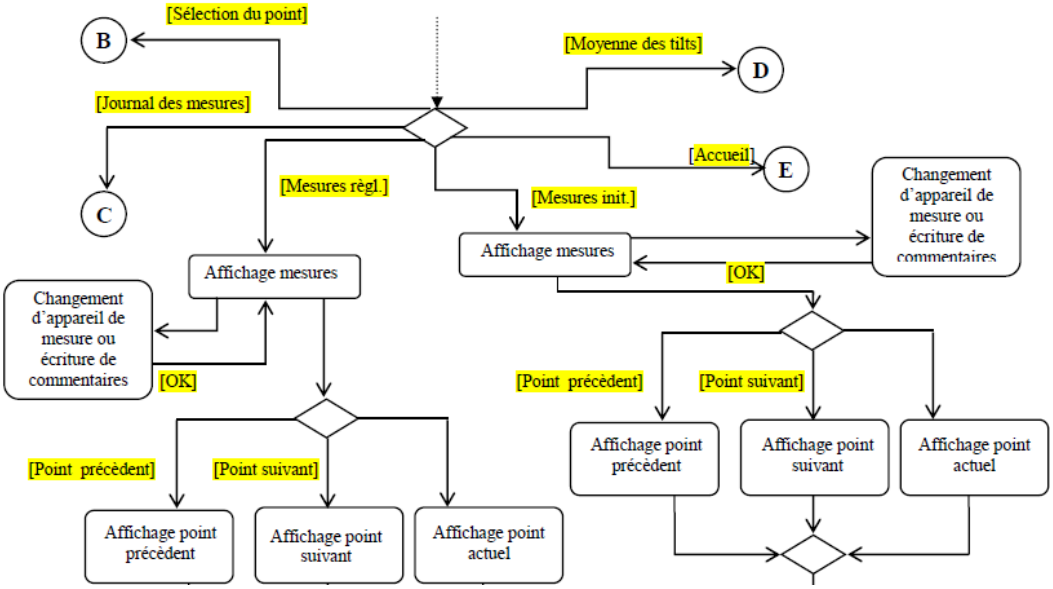


Figure 23 : Extrait du diagramme d'activité - Module de tilt

L'appui sur l'onglet 'Mesure règl.' (correspondant à une prise de mesure après réglage) ou 'Mesure init.' (correspondant à une prise de mesure avant réglage) déclenche le début de l'opération de prise de mesure et valide le point précédemment sélectionné. Il est possible

de laisser des commentaires ou de sélectionner un appareil de mesure différent de celui sélectionné lors du réglage des références. A la fin de la prise de mesures (2 mesures), le logiciel calcul les mesures moyennes et l'écart avec le théorique. L'appui sur le bouton enregistrement déclenche l'écriture des mesures sur le fichier .dat.

Enfin l'utilisateur à la possibilité d'afficher la dernière prise de mesure enregistrée dans l'onglet 'Journal'. L'onglet 'Moyenne des tilts' calcule la moyenne de 6 mesures de tilt au maximum. La valeur moyenne calculée est ensuite enregistrée dans la cellule qui était sélectionnée précédemment (dans l'onglet 'Mesures règl.' ou 'Mesure init.').

V.5.2 La conception détaillée – Module de tilt

Le diagramme de séquences pour le module de tilt se trouve en annexe 13, il s'agit du diagramme pour la validation des références.

V.5.2.1 Diagramme de séquences – Validation des références

Le diagramme de séquences en annexe 13 détaille les interactions entre les différentes classes lors de la validation de toutes les références renseignées par l'utilisateur. Quatre classes sont utilisées, TFrmReference, TReference, TInstrument et TPointsGeode. TFrmReference est la classe qui gère l'interface graphique pour le renseignement des références. Selon les informations saisies par l'utilisateur, 4 variables globales (utilisées dans le logiciel de façon permanente) vont être renseignées. gRef et gRefRegl qui sont des objets de type 'TReference' et qui intègrent les informations sur le nom de l'équipe et le numéro de l'opération. gAppa qui est un objet de type 'TInstrument' qui intègre les informations sur l'instrument choisi. gPointsGeode qui est un objet de type TPointsGeode qui hérite des classes TPoints et TAimants et qui intègre le fichier théorique sous forme de deux listes

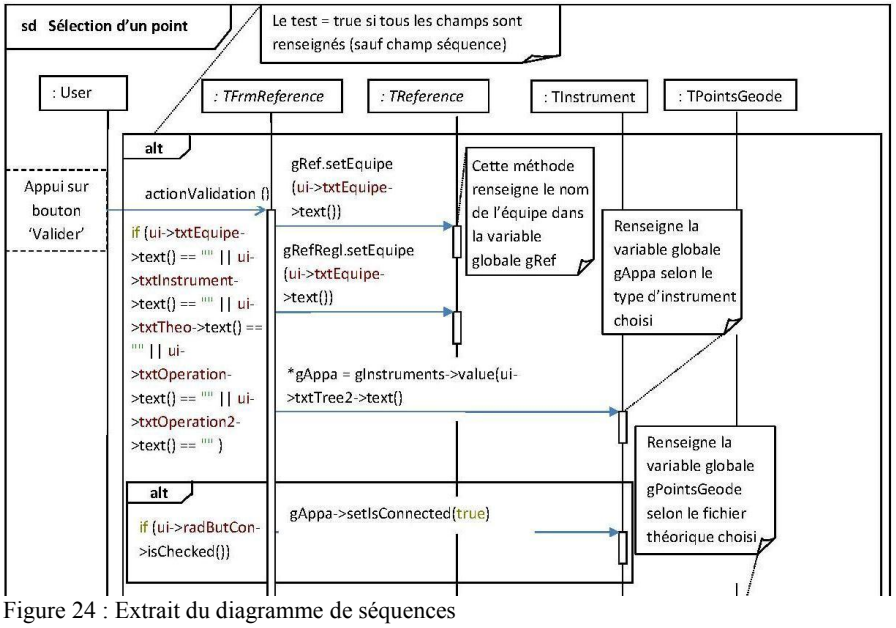


Figure 24 : Extrait du diagramme de séquences

dynamiques (QVector[13]) : une pour la liste des points et une, pour la liste des aimants. Cette variable servira de base pour les mesures qui seront réalisées par la suite.

V.5.3 Les diagrammes d'activité – Module de nivellement

Les diagrammes d'activité pour le module nivellement se trouvent en annexe 14, le surlignage correspond aux interactions de l'utilisateur avec l'interface graphique, les autres interactions correspondant aux interactions entre les méthodes.

Outre la validation des références qui va déclencher l'appel de diverses méthodes, l'activité principale de ce module est la gestion du choix des points à mesurer.

En effet, l'utilisateur peut rajouter des points à mesurer soit à partir d'une liste prédéfinie (créée à partir du fichier théorique), soit les rajouter manuellement ou encore les choisir à partir de séquences prédéfinies (créées à partir du fichier de séquence). L'utilisateur peut mixer ces trois modes de choix, par exemple choisir une séquence puis rajouter un point à partir de la liste prédéfinie et enfin créer un point manuellement. Dans cette liste de points ainsi créée on peut aussi changer l'ordre des points, trier la liste ou encore supprimer un point.

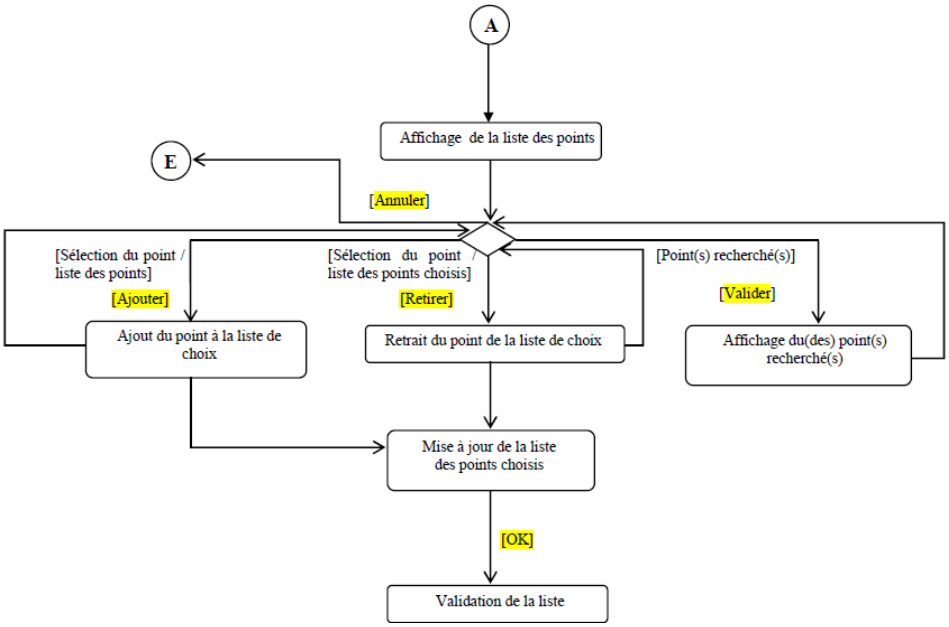


Figure 25 : Extrait du diagramme d'activité - Module de nivellement

L'appui sur l'onglet mesure déclenche le début de l'opération de prise de mesure et valide la liste des points précédemment sélectionnées. Les mesures sont dès lors inscrites

[13] La classe QVector est une classe template qui fournit un tableau dynamique.

manuellement ou en automatique si l'appareil de mesure est connecté. Il est possible de sélectionner une mire différente par rapport à celle sélectionnée au départ et de laisser des commentaires sur chaque point mesuré.

A la fin de la prise de mesure, l'appui sur le bouton enregistrement déclenche l'écriture des mesures sur le fichier .dat et, s'il s'agissait d'une prise de mesure 'aller', l'écriture d'un fichier 'retour.txt' destiné à la gestion des mesure 'retour'. Par ailleurs, si l'utilisateur a choisi de mesurer une station (choix d'une séquence), la station suivante s'affiche automatiquement pour une nouvelle prise de mesure.

Lorsque l'on sélectionne les mesures 'retour', le logiciel va charger le fichier 'retour.txt'et l'utilisateur choisit dès lors une station de départ à mesurer : la première qui s'affiche correspond à la dernière mesure 'aller' effectuée.

L'appui sur l'onglet mesure déclenche le début de l'opération de prise de mesure selon la station précédemment sélectionnée. Les mesures sont dès lors inscrites manuellement ou en automatique si l'appareil de mesure est connecté. Dès que la saisie est réalisée pour chaque point, il s'affiche l'écart avec la mesure 'aller'.

A la fin de la prise de mesure, l'appui sur le bouton enregistrement déclenche le calcul de l'emq (erreur moyenne quadratique sur les écarts de mesure entre l'aller et le retour). Si l'emq est supérieur à 8, le logiciel propose à l'utilisateur une reprise de la station, c'est-à-dire une nouvelle prise de mesure sur la station actuelle. Quel que soit le choix de l'utilisateur, l'écriture de cette station retour est enregistrée dans le fichier .dat. La reprise sera elle aussi enregistrée dans le fichier .dat après saisie. Par ailleurs, la station suivante selon les mesures prises à l'aller s'affiche automatiquement pour une nouvelle prise de mesures.

Enfin l'utilisateur a la possibilité d'afficher la dernière prise de mesure enregistrée dans l'onglet 'mesures précédentes'.

V.5.4 La conception détaillée – Module de nivellement

Les diagrammes de séquences pour le module de nivellement se trouvent en annexe 15 et 16.

V.5.4.1 Diagramme de séquences – Sélection des points

Le diagramme de séquences en annexe 15 détaille les interactions entre les différentes classes lors de la sélection des points à mesurer. Trois classes sont utilisées, TFrmNiv, TANiv et TListePoint. TFrmNiv est la classe qui gère l'interface graphique pour les

mesures de nivellement. TANiv est une classe abstraite qui est héritée par TFrmNiv pour la manipulation des modèles de données utilisés. Par exemple la méthode 'fillList' permet d'ajouter au modèle passé en argument la liste des points provenant du fichier théorique. Ce modèle sera ensuite utilisé par la GUI (un objet Qt de type QTableView[14]) pour afficher la liste des points à sélectionner dans l'onglet 'Choix Points'. L'appui sur le bouton valider copie la liste des points sélectionnés (c.-à-d. ajouté au 'modelSelect') dans le modèle 'modelChoix' et grâce à la méthode 'encodageListeView', la liste des points dans le modèle est affichée sur quatre colonne car la méthode a pour but de séparer les différents éléments composants un point (Zone, Classe, Numéro et Type). Le modèle 'modelChoix' est affiché dans un QTableView dans l'onglet 'Points'. Le cadre 'alt' du diagramme correspond à un opérateur conditionnel. Ici la condition est vraie si le nombre de points sélectionné est supérieur à 1 (correspondant à une station minimale) et dans ce cas on affiche le modèle 'modelChoix' à la vue correspondante dans l'onglet 'Points'.

V.5.4.2 Diagramme de séquences – Ecriture et enregistrement des mesures

Le deuxième diagramme de séquence figurant en annexe 16 détaille comment les mesures vont être écrites puis enregistrées dans le fichier de sortie .dat. Les cadres 'alt' dans les diagrammes correspond aux choix réalisés par l'utilisateur comme sélectionner un type de mesure 'aller' ou 'retour' ou un appareil connecté ou non. La méthode 'startEditMes' qui revient souvent est appelée à chaque fois que l'utilisateur touche le tableau de saisie des mesures et appelle le clavier virtuel pour la saisie (sauf si l'utilisateur utilise un appareil connecté). De plus, si l'utilisateur touche un élément correspondant à la colonne liée à l'affichage de la mire utilisée, cela déclenche l'ouverture des zones permettant l'enregistrement des commentaires et une liste de choix (QCombobox[15]) permettant de sélectionner une autre mire. La sélection de l'option 'retour' déclenche l'appel de la méthode 'open' sur la variable globale 'gSequRetour' qui correspond à un objet 'TNivStations' et chargera le fichier 'retour.txt' créé ou complété lors des mesures 'aller'. Les stations créées dans la variable 'gSequRetour' sont inversées afin que les géomètres réalisent les mesures dans le sens inverse.

---

[14] La classe QTableView fournit une implémentation par défaut d'un modèle/vue sous la forme d'une vue en table.
[15] Le widget QComboBox est un bouton combiné d'une liste déroulante.

Pour les mesures en automatique, l'objet 'TCommNiv' créé lors de l'instanciation de TFrmNiv va subir une dérivation de type afin que les méthodes utilisées pour les communications correspondent avec l'appareil choisi lors du réglage des références. Par exemple, si l'utilisateur a choisi un niveau DNA03, l'objet 'TCommNiv' deviendra un objet de type 'TCommDNA'. Ceci est possible lorsque l'objet dérivé est une classe mère et que les méthodes utilisées sont déclarées 'virtuelles'. Pour effectuer les mesures avec connexion Bluetooth, l'utilisateur doit tout d'abord initialiser la communication par le biais du bouton 'Init', ce qui provoque l'appel de la méthode 'actionInit' qui elle-même va appeler la méthode 'Connect' puis 'Init' de l'objet 'TCommNiv'. L'appui sur le bouton 'Mesurer' va quand à lui appeler la méthode 'mesPoint' qui va appeler la méthode 'Mes'. Si l'utilisateur avait sélectionné une cellule pouvant recevoir une mesure (colonne n°1 dans le diagramme), le modèle gérant les données du tableau des mesures est mis à jour pour enregistrer la valeur mesurer.

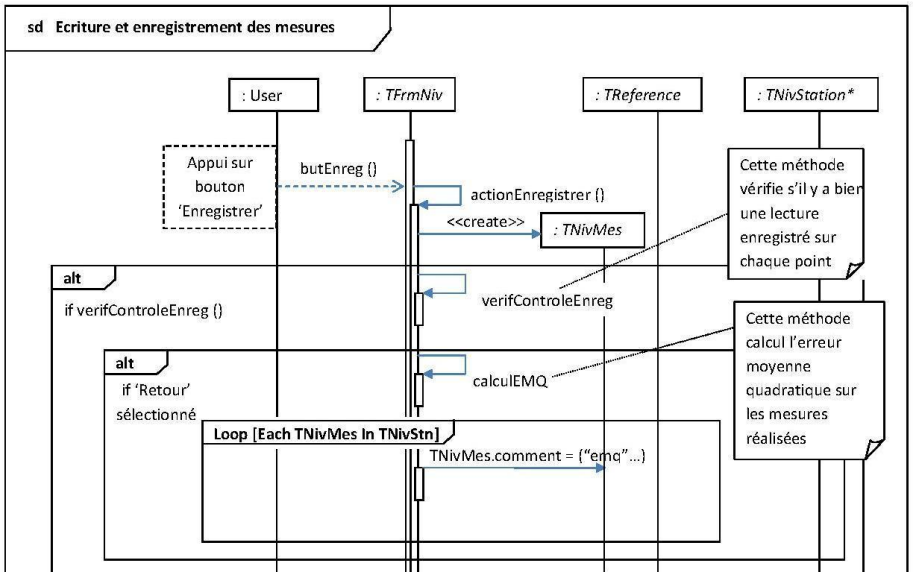


Figure 26 : Extrait du diagramme de séquences

Dès que l'utilisateur a effectué toutes ces mesures, il appui sur le bouton 'Enregistrer' qui appelle la méthode 'actionEnregistrer'. Il y a dès lors l'appel de la fonction 'verifControlEnreg' qui effectue des vérifications sur les mesures réalisées puis le calcul de l'erreur quadratique moyenne (pour les mesures retour). Ensuite, le fichier de sortie .dat est créé ou complété par les méthodes 'ecritGeode' et le fichier 'retour.txt' est créé s'il s'agit de mesures 'aller'. Enfin, si l'emq est hors tolérance, il est proposé à l'utilisateur la reprise de la station. Si l'utilisateur valide la reprise, la méthode 'actionReprise' est appelée qui elle appelle la méthode 'preparerStation'.

V.5.5 Les diagrammes d'activité – Module d'écartométrie

Les diagrammes d'activité pour le module d'écartométrie se trouvent en annexe 17, le surlignage correspond aux interactions de l'utilisateur avec l'interface graphique, les autres interactions correspondant aux interactions entre les méthodes.

Outre la validation des références qui va déclencher l'appel de diverses méthodes, l'activité principale de ce module est la gestion du choix des points à mesurer, et, à ce titre, l'activité fonctionne de la même manière que le module de nivellement.

L'appui sur l'onglet mesure déclenche le début de l'opération de prise de mesure et valide la liste des points précédemment sélectionnés. Les mesures sont dès lors inscrites manuellement ou en automatique si l'appareil de mesure est connecté. Il est possible de sélectionner un appareil différent par rapport à celui sélectionné au départ et de laisser des commentaires sur chaque point mesuré.

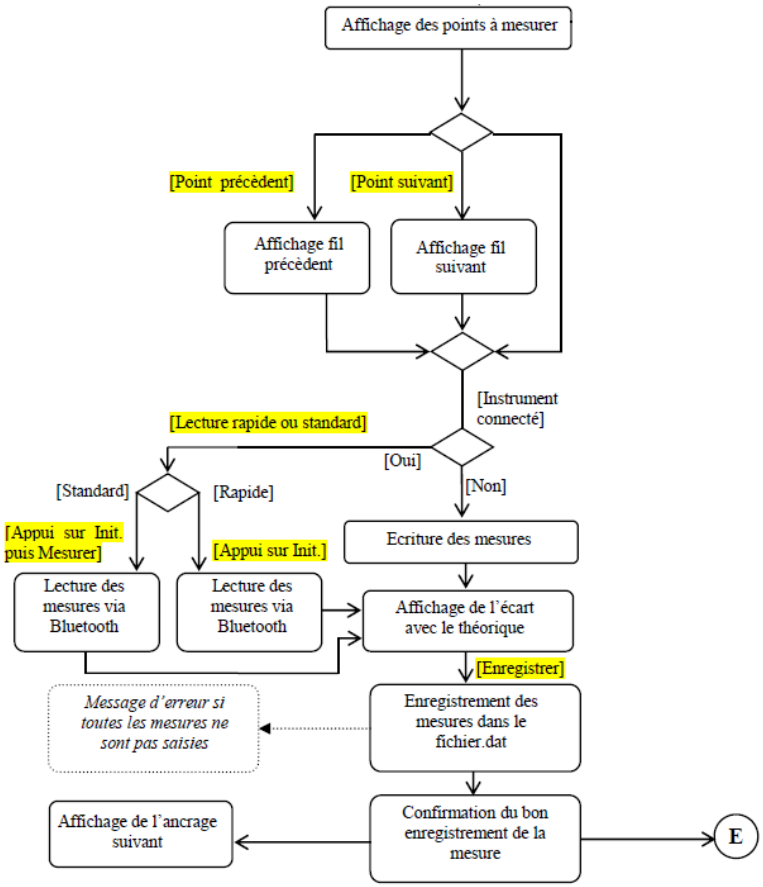


Figure 27 : Extrait du diagramme d'activité - Module d'écartométrie

A la fin de la prise de mesure, l'appui sur le bouton enregistrement déclenche l'écriture des mesures sur le fichier .dat. Par ailleurs, si l'utilisateur a choisi de mesurer un 'fil' (choix d'une séquence), le fil suivant s'affiche automatiquement pour une nouvelle prise de mesure.

Les mesures en automatique disposent de deux modes, 'Rapide' et 'Standard'. Le fonctionnement de ces deux modes sont détaillés dans la partie suivante traitant de la conception détaillée.

Enfin l'utilisateur à la possibilité d'afficher la dernière prise de mesure enregistré dans l'onglet 'mesures précédentes'

V.5.6 La conception détaillée – Module d'écartométrie

Le diagramme de séquences pour le module d'écartométrie se trouve en annexe 18, il s'agit du diagramme pour la sélection d'une séquence.

V.5.6.1 Diagramme de séquences – Sélection d'une séquence

Le diagramme de séquences en annexe 18 détaille les interactions entre les différentes classes lors de la sélection d'une séquence (ou fil) à mesurer. Quatre classes sont utilisées, TFrmEcarto, TAEcarto, TFil et TFilMes. TFrmEcarto est la classe qui gère l'interface graphique pour les mesures d'écartométrie. TAEcarto est une classe abstraite qui est héritée par TFrmNiv pour la manipulation des modèles de données utilisées. Quand l'utilisateur appui sur le bouton 'Séquence', cela déclenche l'appel de la méthode 'actionSequence' dont le but principal est de compléter le modèle 'modelSequ' avec les données provenant de la variable globale 'gSeqFils' où sont chargées les différentes séquence de fils à mesurer. Grâce à l'utilisation d'une boucle (loop), les différents fils sont chargés et l'on garde uniquement les deux premiers éléments de chaque qui correspondent aux deux ancrages. Ces deux ancrages sont ensuite ajoutés au modèle 'modelSequ'. Le modèle affiche maintenant les deux ancrages pour chaque séquence pour que l'utilisateur fasse le choix d'un fil à mesurer.

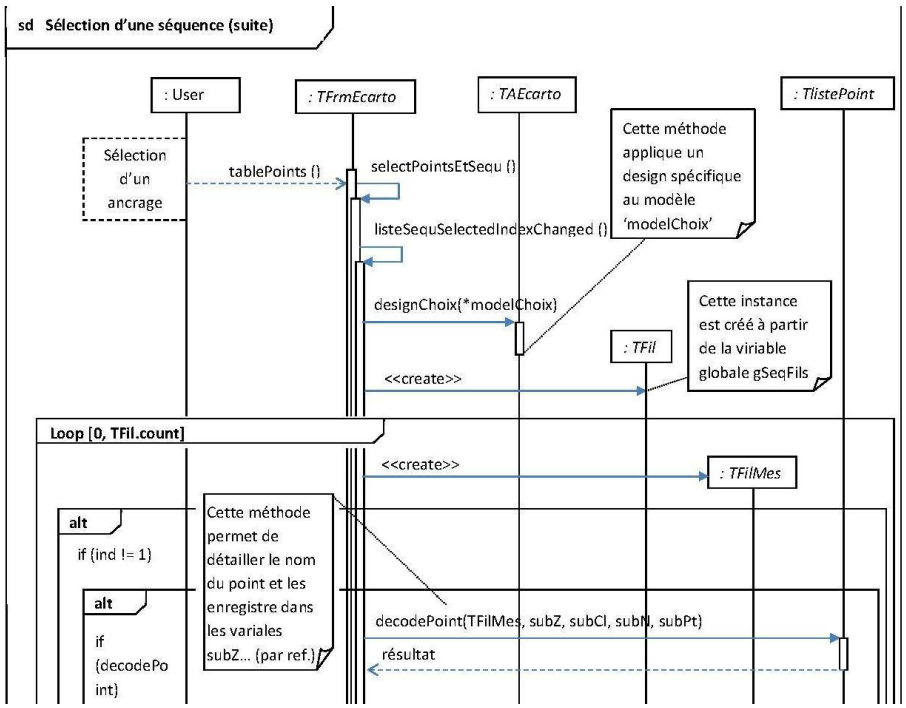


Figure 28 : Extrait du diagramme de séquences

V.6 La réécriture du logiciel

La réécriture du logiciel a été réalisé avec pour base les diagrammes évoqués lors du chapitre précédent et j'ai essentiellement travaillé sur la version Android. De nombreuses

classes dans le logiciel original écrit en VB.Net définissait les objets utilisés (Point, Aimant, NivMes,…) et la réécriture de ces classes ne fut pas un problème, le langage C++ étant lui aussi un langage objet, il a suffit de convertir les classes d'un langage à un autre. Le point le plus compliqué du projet a été de travailler avec la version 5.1 de Qt qui apportait un support 'préliminaire' d'iOS et Android. De ce fait, j'ai rencontré plusieurs difficultés pour adapter l'interface graphique du logiciel à l'univers mobile Android. Je détaille ces difficultés dans cette partie et complète la description du travail réalisé par la méthode employée pour réaliser les tests. Je commente les optimisations réalisées sur le nouveau logiciel par rapport à l'ancien et je termine par les tests réalisés par les utilisateurs et les corrections que j'ai réalisées.

V.6.1 Difficultés

V.6.1.1 Accès au système de fichier

Normalement, la classe QFileSystemModel apporte un modèle de donnée pour le système de fichier local. Cette classe devait être utilisée pour l'accès au système de fichier pour le choix du fichier théorique et de séquence. Malheureusement elle n'était pas supportée correctement par la version Android de Qt. Afin de solutionner ce problème, j'ai utilisé la classe QDirModel qui fournie elle aussi un accès au système de fichier.
Voici le code utilisé permettant l'usage de la classe QDirModel dans un QTreeView qui est une classe permettant l'implémentation d'une vue en arbre :

```
void TFrmReference::chargeTreeTheo()
{
    QDirModel *modelFich = new QDirModel;
    QStringList filters;
    filters << "*.dat";
    modelFich->setNameFilters(filters);
    modelFich->setFilter(QDir::AllDirs | QDir::NoDotAndDotDot | QDir::Files);

    ui->treeViewTheo->setModel(modelFich);
    ui->treeViewTheo->setColumnHidden(1, true);
    ui->treeViewTheo->setColumnHidden(2, true);
    ui->treeViewTheo->setColumnHidden(3, true);
    ui->treeViewTheo->setRootIndex(modelFich->index(gRep));
    ui->treeViewTheo->setAnimated(true);
    ui->treeViewTheo->resizeColumnToContents(0);
}
```

Figure 29 : Implémentation QDirModel

Tout d'abord l'on crée une instanciation de QDirModel ayant pour nom 'modelFich'. Puis j'ai voulu appliquer un filtre au modèle afin qu'il affiche uniquement les fichiers portant l'extension '.dat' car ce sont uniquement ces types de fichiers qui sont utilisés par le

programme (fichiers théorique et séquence). Ce filtrage est possible par la fonction 'setNameFilters' qui prend en argument un objet de type 'QStringList'. Ensuite la fonction permet de réaliser des filtres sur les répertoires à afficher. Ici, j'ai choisi tous les répertoires (AllDirs), les fichiers (Files) et fait le choix de ne pas afficher les caractères spéciaux '.' et '..' (NotDotAndDotDot).

Le modèle est alors associé à la vue (QTreeView) par la fonction 'setModel'.

V.6.1.2 Adaptation de l'interface graphique à un affichage tactile

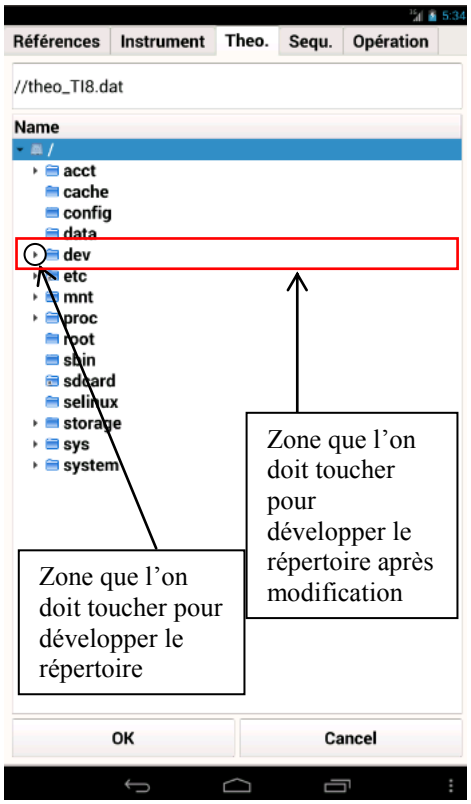


Figure 30 : Ecran de sélection d'un fichier théorique

La vue utilisée précédemment (QTreeView) a permis d'afficher le système de fichier local. Cependant, le support partiel des interfaces tactiles par Qt est ici illustré par l'usage de cette vue.

En effet, comme c'est évoqué dans la figure ci-contre, la zone que l'on doit sélectionner pour développer un répertoire est ergonomiquement destinée à un appareil fonctionnant à l'aide d'une souris, cette zone étant très difficilement sélectionnable sur un terminal tactile. Pour solutionner le problème, j'ai tout d'abord réalisé une connexion entre un signal et un slot (voir figure 31).

Le signal provient de la vue (QTreeView) qui, lorsque un élément est sélectionné (clicked) appelle le slot 'expandTheo()'(voir figure 32).

```
connect(ui->treeViewTheo, SIGNAL(clicked(QModelIndex)), this, SLOT(expandTheo()));
```

Figure 31 : Connexion signal/slot pour adaptation tactile

Le slot récupère tout d'abord l'index de l'élément sélectionné (selection->currentIndex), puis grâce à la fonction 'expand' du QTreeView développe le répertoire sélectionné.

```
void  TFrmReference::expandTheo()
{
    QItemSelectionModel *selection = ui->treeViewTheo->selectionModel();
    QModelIndex indexElementSelectionne = selection->currentIndex();
    ui->treeViewTheo->expand(indexElementSelectionne);
}
```

Figure 32 : Slot pour adaptation tactile

V.6.1.3 Développement des classes permettant les mesures en automatique

Pour les mesures de nivellement et d'écartométrie, l'utilisateur du logiciel a la possibilité d'utiliser des appareils de mesure connectés. Les appareils de mesures pour le nivellement (niveau) utilisent un adaptateur RS232 -> Bluetooth. Ainsi, via la connexion Bluetooth de l'appareil mobile, il est possible de réaliser des mesures automatiques sans fil. Pour l'écartométrie, la connexion Bluetooth se fait directement.

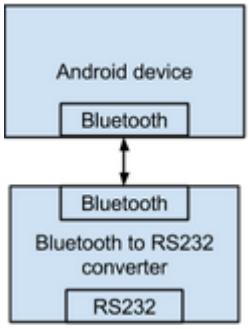


Figure 33 : Adaptateur Bluetooth/RS232

Afin, de réaliser le codage des classes permettant la communication entre le terminal mobile et l'appareil de mesures, j'ai dû utiliser un logiciel permettant de voir les informations entrée/sortie transitant dans la communication RS232. En annexe 19 est présenté le logiciel utilisé, Free Serial Port Monitor, qui permet donc d'intercepter les données échangées : il m'a permis d'ajuster les réglages dit 'TimeOut'[16]. Dans l'exemple de code ci-dessus, les réglages TimeOut sont définis dans la méthode 'PortSendData' qui prend en argument dans l'ordre l'objet 'serialport' à utiliser, la requête à envoyer, la variable de type string qui va récupérer la réponse de la requête, le timeout pour l'envoi de la requête et le timeout pour la lecture de la réponse. On peut voir ici que les réglages adoptés sont 2000 millisecondes pour les TimeOut d'envoi et de réponse. Ces réglages ont dû être adaptés à chaque appareil de mesure car le temps d'attente pour que

```
QString TCommDINI::SendMes(QSerialPort &port)
{
    QByteArray reponse;

    try
    {
        QThread::msleep(50);
        TSerialPort serial;
        serial.PortSendData(port, "FML" + crlf, reponse, 2000, 2000);
        QThread::msleep(2000);

        mMeasuring = true;

        QString mesure(reponse);

        return mesure;
    }
```

Figure 34 : Méthode 'SendMes' de la classe TCommDINI

[16] Délais d'attente d'une lecture ou d'écriture sur le port série

l'appareil soit prêt à recevoir une requête ou le temps d'attente pour la réalisation de la mesure et de l'envoi de la réponse sur le port série était spécifique à chaque appareil.

V.6.2 L'adaptation de l'interface graphique à différent terminal

Le nouveau logiciel PFB développé devra fonctionner sans modification majeure sur la majeure partie des terminaux mobiles présents sur le marché, autant les smartphones que les tablettes tactilles. Aussi, cela pose un problème quand à l'adaptation de l'interface graphique à la taille de l'écran. Si l'on place chaque élément composant l'interface de manière absolue, celle-ci ne s'adaptera pas aux différentes définitions et taille d'écran. Pour solutionner ce problème, l'outil Qt Designer qui rend possible la création rapide d'interfaces graphiques et surtout l'utilisation de layout qui permet aux widgets de s'adapter aux différentes tailles d'écran. En annexe 20 est présenté Qt Designer où l'on a le choix entre plusieurs types de layout : les layouts horizontaux, verticaux, de grilles et de formulaires. Pour réaliser mon interface graphique, j'ai combiné tous les types de layouts sauf celui du formulaire. Si j'avais besoin d'un alignement horizontal de boutons, j'utilisais un layout horizontal pour les intégrer et j'intégrais ce même layout dans un layout de grille. Ainsi, en définissant une taille minumum et maximum pour chaque widget ainsi qu'une politique d'extension (taille fixe, étendu en horizontal et/ou vertical, minimum ou maximum), tous les widgets adaptaient leurs tailles par rapport à la taille et la définition de l'écran du terminal mobile et aussi par rapport à l'orientation de l'écran. En effet, grâce à l'accéléromètre des terminaux mobiles, l'écran s'adapte automatiquement en temps réel en mode portrait ou paysage.

L'annexe 21 présente des exemples d'affichage sur différents terminaux comportant une taille d'écran et une définition diférente.

V.6.3 Les tests

Le planning que j'ai établi intègre des phases de tests unitaires et d'intégration. Contrairement à ce qui est réalisé dans le cadre d'un développement normal d'une application, il est conseillé d'écrire les tests avant la fonction à tester. Les tests unitaires servent à vérifier le fonctionnement d'une partie précise d'un logiciel et les tests d'intégration consistent à vérifier, en cas de développement d'un logiciel par plusieurs développeurs, le fonctionnement de l'agrégat des différentes parties de celui-ci ensemble. Etant donné que j'étais tout seul pour développer le nouveau logiciel, l'on ne parle plus de tests unitaires et d'intégration mais plutôt d'une intégration continue (fusion des tests unitaire et d'intégration).

Lors de mon développement, j'ai pris pour base de tests la version originale et fonctionnelle de PFB sous VB.Net car celle-ci était éprouvée par des années d'utilisation est était une base fiable pour vérifier le bon comportement de mon nouveau logiciel sous Android.

Ainsi, à chaque nouvelle méthode écrite sous Qt, j'ai réalisé les tests unitaires en comparant le comportement de la version de PFB sous VB.Net et la version Qt, en annexe 21 sont présentés les écrans de débugage sous Visual Studio 2005 (ancienne version de PFB) et Qt Créator 2.8.1 (nouvelle version). A chaque point clé du fonctionnement d'une méthode testée, j'ai vérifié les valeurs de chacune des variables (arguments, variables de classes, variable à portée limitée à la méthode, valeur de retour de la méthode) et le comportement des deux interfaces graphiques.

Enfin, pour valider le bon fonctionnement du logiciel par un utilisateur, à chaque module terminé (Tilt, Nivellement et Ecartométrie), le logiciel fut installé sur une tablette tactile Android de test afin que des tests en condition réelle soient réalisés.

V.6.4 Les retours utilisateurs et corrections réalisées

Les tests réalisés par les utilisateurs sont les plus importants car ce sont eux qui vont tester le logiciel conformément à ce qu'implique leur métier. Des tests par des développeurs restent limités dans leur exhaustivité car ils n'ont pas forcément en tête tous les besoins 'métier'. Dans cette partie je détaille les corrections ainsi que les optimisations principales réalisées sur le logiciel.

V.6.4.1 Corrections réalisées

Les corrections réalisées ont été surtout liées au comportement de l'interface graphique et plus rarement à des problèmes liés à des méthodes donnant des résultats incorrects.

Pour le module nivellement, un des problèmes principaux fut la méthode qui calculait l'emq (erreur quadratique moyenne). Pour les mêmes mesures, je n'arrivais pas à obtenir le même résultat entre l'ancienne version du logiciel et la nouvelle. Pourtant la formule utilisée pour le calcul était la même. C'était tout simplement dû à la priorité des opérateurs qui différait entre le langage VB.Net et C++.

```
                diff(n) = nivAi.Lect.Value + nivAi.Rallonge * 100 - nivRi.Lect.Value -
nivRi.Rallonge * 100
```

Figure 35 : Portion de code - Calcul de l'emq - VB.Net

```
            diff.push_back(nivAi->lect() + (nivAi->rallonge() * 100) - nivRi->lect() -
(nivRi->rallonge() * 100));
```

Figure 36 : Portion de code - Calcul de l'emq - C++

Après avoir rétabli la bonne priorité des opérateurs grâce à l'emploi de parenthèses, le problème a été résolu.

Un exemple de correction concernant l'interface graphique fut réalisé dans le module de nivellement : lorsque l'utilisateur n'a pas mesuré tous les points de la station, il doit quand même pouvoir enregistrer sa prise de mesure. En annexe 22 se trouve la solution à ce problème. Tout d'abord, par la boucle 'for' on balaye toutes les mesures enregistrées (objet nivStn). Dans la première condition, si une des lectures est égale à -99999999 (les lectures sont initialisées à cette valeur car elles ne peuvent pas être initialisées à 0, car 0 est une valeur de mesure valide), c'est-à-dire qu'il n'y a pas eu de mesure sur ce point, cela déclenche l'ouverture d'une boite de dialogue qui prévient qu'il n'y a pas de lecture sur le point concerné. Si l'utilisateur appui sur 'OK', la valeur de retour de la boite de dialogue est égale à 1024. Dans la deuxième condition, si l'utilisateur à appuyer sur 'OK' et si aucune mesure n'a été saisie sur le point, le point est supprimé de la station mesurée. Dans la troisième condition, une fois que l'on est sorti de la boucle for, si l'utilisateur a appuyé sur 'Cancel' (valeur différente de 1024), la méthode retourne la valeur 'false' ce qui invalide la procédure d'enregistrement. Sinon, la valeur 'true' est renvoyée et l'enregistrement des mesures aura bien lieu.

V.6.4.2 Optimisations réalisées

Pour l'interface graphique, une des optimisations était liée à la saisie rapide des mesures. J'ai réalisé cette amélioration principalement par la connexion signal/slot suivante :

connect(modelMes,SIGNAL(itemChanged(QStandardItem*)), this, SLOT(calculMes()));

Cette connexion détecte si un élément a été modifié dans le modèle 'modelMes' qui est utilisé sur la vue où l'on réalise les mesures. Si c'est le cas, la méthode calculMes est appelée. Cette méthode a pour but de calculer les écarts de mesure avec les stations 'Aller'. Je l'ai aussi utilisée pour que la cellule sélectionnée soit la suivante et aussi que le clavier virtuel apparaisse automatiquement, et ainsi de suite jusqu'à la dernière mesure soit réalisée.

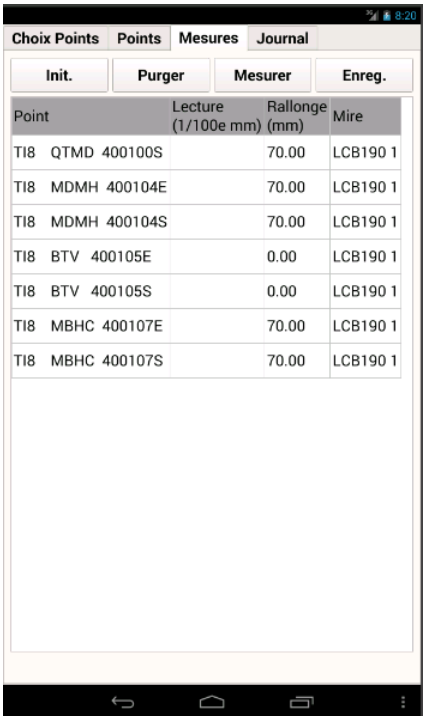


Figure 37 : Nivellement - Saisie des mesures

Une autre optimisation est liée à l'affichage du journal des dernières mesures réalisées. Sur l'ancienne version du logiciel, on pouvait uniquement consulter la dernière mesure réalisée : sur la nouvelle version, on peut consulter toutes les mesures depuis le début des prises de mesures sur le même module. Dans un premier temps, quand l'utilisateur a cliqué sur l'enregistrement des mesures, les données du modèle sont enregistrées dans des listes dynamiques (voir figure ci-dessous). Ces mêmes listes dynamiques sont enregistrées dans une autre liste (variable de classe f_ListeNivRecord). Pour l'affichage du journal, la méthode 'afficheJournal' est appelée, elle est présentée en l'annexe 23. Cette méthode est appelée à chaque fois que l'on clique sur l'onglet 'Journal' de l'interface graphique du logiciel et aussi à chaque fois que l'on veut afficher des mesures réalisées précédemment ou ultérieurement. La méthode utilise les variables de classes 'indSeqJournAffich' et 'indSeqJourn' pour gérer l'ordre d'affichage du journal. Pour afficher le journal, des listes dynamiques représentant les différentes colonnes à afficher (nomPoints, lectures,…) sont prélevées de la variable de classe 'f_ListeNivRecord'. Puis selon le nombre de ligne dans la liste dynamique 'nomPoints', le modèle de données utilisé 'modelJour' pour l'affichage du journal, est mis à jour avec les données présentes dans chaque liste dynamique.

```
QVector<QString> nomPoints;
QVector<QString> lectures;
QVector<QString> rallonges;
QVector<QString> ecart;
QVector<QString> mire;

int nb = modelMes->rowCount();

for (int i = 1;  i < nb; i++)
{
    nomPoints.append(modelMes->item(i, 0)->text());
    lectures.append(modelMes->item(i, 1)->text());
    rallonges.append(modelMes->item(i, 2)->text());
    ecart.append(modelMes->item(i, 3)->text());
    mire.append(modelMes->item(i, 4)->text());
}

f_ListeNivRecord->append(nomPoints);
f_ListeNivRecord->append(lectures);
f_ListeNivRecord->append(rallonges);
f_ListeNivRecord->append(ecart);
f_ListeNivRecord->append(mire);
```

Figure 38 : Nivellement - Copie des mesures au journal

V.6.5 Bilan de la réécriture

Une fois tous les tests réalisés et le logiciel prêt à être déployer je peux dresser le bilan suivant :

- Le point négatif est le non support du module QtSerialPort sur la version Android de Qt, ce qu'il ne m'a pas permis de tester le mode automatique des opérations de mesures avec l'appareil mobile. Le support du module pour Android n'est pas annoncé sur la version 5.2 de Qt, ce qui repousse l'utilisation en automatique du logiciel.
- Le planning annoncé a finalement été respecté et, hormis le problème avec le support du module QtSerialPort, le logiciel est prêt à être utilisé.
- Les classes permettant la communication avec les appareils de mesure ont été testées sous Windows. Le code Android est identique, les fonctions de communications non supportées pour l'instant ont été mises en commentaires afin qu'elles soient activées rapidement en cas de support du module QtSerialPort dans la version Android de Qt.
- La version iOS n'a pas été testée à ce jour. Normalement, elle doit être identique à la version Android hormis un point important : iOS ne permet pas un accès sur le système de fichier et l'insertion et la récupération des fichiers n'est pas possible avec le code existant. La solution consisterais à créer une zone d'échange de fichier au sein même du logiciel et de récupérer ou d'insérer des fichiers par le logiciel Apple gérant la synchronisation des données avec les appareils iOS : iTunes.
- Le point positif et des plus appréciables après les nombreuses heures passées à réécrire le logiciel et tester son comportement est que celui-ci sera mis en production prochainement car les Pocketpc sur lesquels tourne l'ancienne version arrivent à bout de souffle.

VI Maintenance et évolution

VI.1 La documentation destinée au développeur

Il est communément admis que le coût d'un logiciel est à 80% lié à sa maintenance et seulement 20% à sa conception. C'est pourquoi la documentation liée à ce nouveau logiciel est particulièrement importante car elle permettra au développeur chargé de sa maintenance de gagner du temps et d'assurer une maintenance aisée.

L'outil utilisé pour le partage de la documentation est un outil déjà utilisé dans de nombreux projets au CERN : Confluence.

Confluence donne un endroit où partager, trouver et interagir avec les informations dont l'équipe à besoin pour travailler et permet de centraliser les informations liées à un développement.

Plusieurs éléments de documentations ont été déposés sur Confluence pour qu'ils soient accessibles par les équipes de la section SU :

- Le guide de déploiement du nouveau logiciel
- Les diagrammes d'activité et de séquences qui ont servi de base à la réécriture
- La documentation du code grâce à l'utilisation de Doxygen qui est un outil qui permet de générer une documentation à partir du code source. Durant mon développement, j'ai apposé des commentaires à mon code selon les standards requis par Doxygen. Ainsi la documentation générée est relativement complète. En annexe 24 est présenté un exemple de commentaire réalisé dans le prototype d'une classe C++ et la documentation générée par Doxygen sur cette même classe.
- Les commentaires sur le code source : ce qu'il reste à améliorer, des conseils sur des fonctionnalités à apporter…

VI.2 La documentation destinée aux utilisateurs

La documentation utilisateur a été créé directement sur Confluence, une copie d'écran de d'une partie de cette documentation est visible en annexe 25. Confluence permet de rendre accessibles facilement et simplement les informations sur l'utilisation du logiciel et de pouvoir réaliser des modifications si besoin directement en ligne et de générer automatiquement par exemple une table des matières avec le lien sur chaque partie ou sous-partie.

Conclusion

Cette expérience de réécriture, selon des critères bien défini, m'a permis de travailler sur plusieurs aspect du domaine de l'informatique. En premier lieu, l'étude de la problématique et l'examen des différentes solutions pour rendre le nouveau logiciel multiplateforme.

Dans un deuxième temps, il a fallu estimer le temps de développement pour la réécriture du logiciel alors que les démarches habituelles pour réaliser ces estimations concernent plutôt des projets nouveaux. La méthode employée s'est avérée bonne car l'ajustement de l'estimation du temps de développement au fur et à mesure de l'avancement a permis de mieux cibler les estimations du temps de développement restant.

Ensuite, l'analyse de l'existant a été rendu difficile par une documentation très peu étoffé, m'a obligé de structurer mon analyse selon les fondamentaux du génie logiciel afin de créer les documents liés à la conception générale et détaillé basé sur le développement à venir en s'inspirant de l'ancien logiciel.

De même, la réalisation des tests a été simplifiée car j'ai suivi le fonctionnement de l'ancien logiciel lors du débogage de mon code, ce qui m'a permis de tenir les délais de développement. Le développement par module m'a aussi permis d'apprendre de mes erreurs sur les précédents modules et m'a permis d'accélérer le développement avec le temps. Néanmoins, pour un projet normal, il convient avant de coder d'écrire les tests afin de garantir un code plus juste et plus fiable.

Ensuite, étant donnée que généralement 80 % du coût du logiciel intervient lors des phases de maintenance et d'évolution, il est obligatoire de réaliser une documentation la plus complète possible afin de garantir une maintenabilité efficace, ce que j'ai réalisé à la fin de mon stage. De plus, les deux objectifs liés à la réécriture ont été atteint, grâce à Qt le logiciel est maintenant multiplateforme et avec l'architecture modèle/vue, les données ont été séparées de la vue.

Pour finir, personnellement, ce stage m'a permis d'enrichir mes compétences en programmation objet et de mettre en application ce que j'ai appris le long de mon cursus au sein du CNAM au niveau de la gestion de projet.

Bibliographie

Alexandre, D., 2012. *Web Storage, le stockage de données côté client.* [En ligne]
Disponible sur: http://fr.clever-age.com/veille/blog/web-storage-le-stockage-de-donnees-cote-client.html
[Accès le 05 12 2013].

Briot, B., 2012. *Développement d'applications mobiles multi-plateforme.* [En ligne]
Disponible sur: http://www.ictjournal.ch/fr-CH/News/2012/03/20/Developpement-dapplications-mobiles-multi-plateforme.aspx?pa=1
[Accès le 05 12 2013].

CERN, 2011. *Worldwide LHC Computing Grid.* [En ligne]
Disponible sur: http://lcg-archive.web.cern.ch/lcg-archive/public/
[Accès le 05 12 2013].

CERN, 2013. *About CERN.* [En ligne]
Disponible sur: http://home.web.cern.ch/about
[Accès le 05 12 2013].

CERN, 2013. *États membres.* [En ligne]
Disponible sur: http://home.web.cern.ch/fr/about/member-states
[Accès le 05 12 2013].

CERN, 2013. *Expériences.* [En ligne]
Disponible sur: http://home.web.cern.ch/fr/about/experiments
[Accès le 05 12 2013].

CERN, 2013. *La structure du CERN.* [En ligne]
Disponible sur: http://home.web.cern.ch/fr/about/structure-cern
[Accès le 05 12 2013].

CERN, 2013. *Le complexe d'accélérateurs.* [En ligne]
Disponible sur: http://home.web.cern.ch/fr/about/accelerators
[Accès le 05 12 2013].

CERN, 2013. *Physique.* [En ligne]
Disponible sur: http://home.web.cern.ch/fr/about/physics
[Accès le 05 12 2013].

CRAZY WS, 2012. *7 frameworks HTML5 pour créer une application web mobile.* [En ligne]
Disponible sur: http://www.crazyws.fr/dev/applis-et-scripts/7-frameworks-html5-pour-creer-une-application-web-mobile-SRZRI.html
[Accès le 05 12 2013].

Digia Oyj, 2013. *Qt digia.* [En ligne]
Disponible sur: http://qt.digia.com/
[Accès le 05 12 2013].

Gbofu, E. D., 2012. *Développement mobile multiplateforme en C et C++ avec MoSync SDK.* [En ligne]
Disponible sur: http://eric-gbofu.developpez.com/tutoriels/cpp/introduction-developpement-mobile-multiplateforme-c-et-cplusplus/
[Accès le 05 12 2013].

Haxe, 2013. *Multiplatform open-source programming language.* [En ligne]
Disponible sur: http://haxe.org/
[Accès le 05 12 2013].

ideo MOBILIS, 2011. *COMPARATIF DES OUTILS DE DÉVELOPPEMENT MULTI-PLATEFORME MOBILE.* [En ligne]
Disponible sur: http://www.journaldunet.com/developpeur/outils/les-outils-de-developpement-multi-plateforme-mobile/rhomobile.shtml
[Accès le 05 12 2013].

ideo MOBILIS, 2011. *Comparatif des solutions de développement multiplateformes mobile.* [En ligne]
Disponible sur: http://www.ideomobilis.com/actualite/comparatif-solutions-developpement-multiplateformes-mobile
[Accès le 05 12 2013].

ideo MOBILIS, 2012. *Comparatif de frameworks HTML-Javascript pour le développement d'applications Web mobiles.* [En ligne]
Disponible sur: http://www.ideomobilis.com/actualite/comparatif-frameworks-html-javascript-developpement-dapplications-web-mobiles
[Accès le 05 12 2013].

Mark Jones, C. P., 2005. *Qualité de la chaine des données Survey et du système de coordonnées CERN,* s.l.: CERN.

MoSync AB, 2013. *Cross-platform SDK and HTML5 tools for mobile app development.* [En ligne]
Disponible sur: http://www.mosync.com/
[Accès le 05 12 2013].

ORACLE, s.d. *Comment obtenir Java pour les dispositifs mobiles ?.* [En ligne]
Disponible sur: http://www.java.com/fr/download/faq/java_mobile.xml
[Accès le 05 12 2013].

Qt Project, 2013. *QtDoc 5.0: Qt 5.0.* [En ligne]
Disponible sur: http://qt-project.org/doc/qt-5.0/qtdoc/index.html
[Accès le 05 12 2013].

WIKIPEDIA, 2013. *Constructive Cost Model.* [En ligne]
Disponible sur: http://fr.wikipedia.org/wiki/Constructive_Cost_Model
[Accès le 05 12 2013].

WIKIPEDIA, 2013. *Java Platform, Micro Edition.* [En ligne]
Disponible sur: http://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition
[Accès le 05 12 2013].

WIKIPEDIA, 2013. *Multiplate-forme.* [En ligne]
Disponible sur: http://fr.wikipedia.org/wiki/Multiplate-forme
[Accès le 05 12 2013].

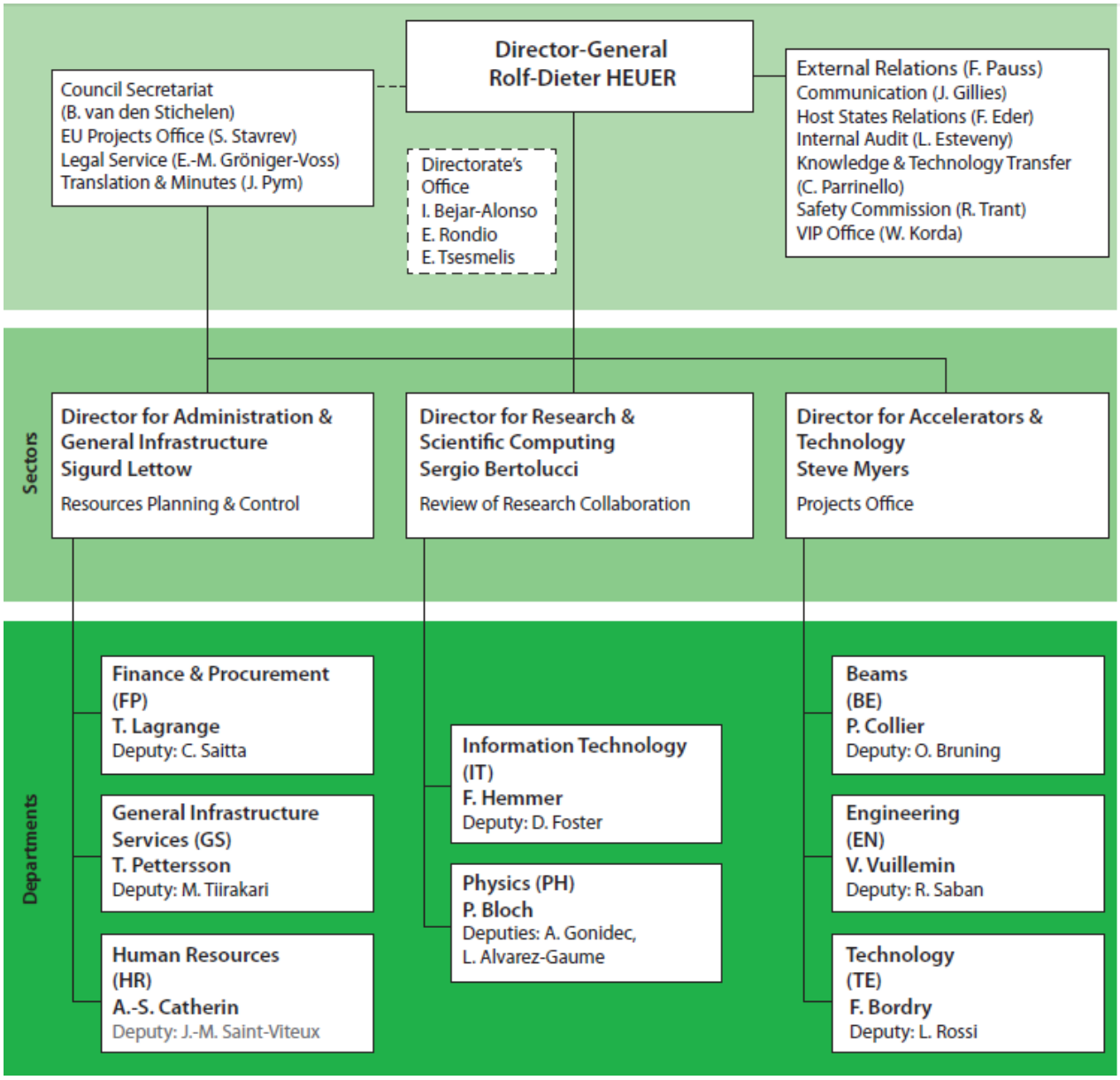
WIKIPEDIA, 2013. *Multiple phone web-based application framework.* [En ligne]
Disponible sur:
http://en.wikipedia.org/wiki/Multiple_phone_web_based_application_framework
[Accès le 05 12 2013].

www.markus-falk.com, s.d. *Mobile Frameworks Comparison Chart.* [En ligne]
Disponible sur: http://www.markus-falk.com/mobile-frameworks-comparison-chart/
[Accès le 05 12 2013].

Table des annexes

Annexe 1 Structure du CERN 83
Annexe 2 Organigramme du département BE 84
Annexe 3 Structure du département ABP 85
Annexe 4 Structure de la section SU 87
Annexe 5 Qt Designer – Ecran d'édition des fenêtres 88
Annexe 6 Points de fonction – Dénombrement des composants 89
Annexe 7 Points de fonction – Addition des points de fonction 91
Annexe 8 Planning prévisionnel de développement – Diagramme de Gantt 92
Annexe 9 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Tilt 93
Annexe 10 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Nivellement 94
Annexe 11 Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Ecartométrie 95
Annexe 12 Diagramme d'activité – Module de tilt 96
Annexe 13 Diagramme de séquence – Module de tilt – Validation des références 99
Annexe 14 Diagramme d'activité – Module de nivellement 100
Annexe 15 Diagramme de séquence – Module de nivellement – Sélection des points 105
Annexe 16 Diagramme de séquence – Module de nivellement – Ecriture et enregistrement des mesures 106
Annexe 17 Diagramme d'activité – Module d'écartométrie 112
Annexe 18 Diagramme de séquence – Module d'écartométrie – Sélection d'une séquence 116
Annexe 19 Free Serial Port Monitor 118
Annexe 20 Utilisation des layouts sous Qt Designer 119
Annexe 21 Tests d'intégration continus : comparaison des deux versions de PFB 121
Annexe 22 Correction du problème de multiples enregistrements 122
Annexe 23 Nivellement - Affichage du journal 123
Annexe 24 Documentation générée par Doxygen 124
Annexe 25 Documentation utilisateur sur Confluence 125

Annexe 1
Structure du CERN



Annexe 2
Organigramme du département BE

BE Beams Department

Structure on 1 March 2013

Roles		
Department Head (DH)	P. Collier	75513
Assistant to the Department Head (ADH)	A. Mackney	74468
Departmental Administrative Officer (DAO)	L. Van Cauter-Tanner	78575
Departmental Planning Officer (DPO)	R. Billen	76904
Departmental Safety Officer (DSO)	M. Tavlet	73717
Departmental Space Manager (DSM)	G. Roy	76954
Departmental Training Officer (DTO)	E. Montesinos	75190
Human Resources Advisor (HRA)	M.-L. Falipou	74701
Industrial Services Coordinators	M. Draper	73348
	E. Jensen	74298

Deputies		
DDH	R. Garoby	72892
DDAO	J. Schueler	72037
DDPO	A. Burns	75296
DDSO	C. Gaignant	62255
DDSM	R. Billen	76904
DDTO	L. Van Cauter-Tanner	78575
DHRA	B. Delille	74469

HDO - Head Departmental Office

P. Collier	Group Leader		75513
R. Garoby	Deputy Group Leader		72892
A. Mackney	*Secretariat*	*Fax: 7673820*	*74468*

ABP - Accelerators & Beams Physics

O. Brüning	Group Leader		79161
G. Arduini	Deputy Group Leader		72944
D. Rivoiron	*Secretariat*	*Fax: 7679145*	*72523*
ABP-CC3	CLIC & CTF3	R. Corsini	78669
ABP-HSL	Hadron Sources & Linacs	R. Scrivens	74601
ABP-ICE	Impedance and Collective effects	E. Metral	72560
ABP-LIS	LHC Injector Synchrotrons	G. Arduini	72944
ABP-LCU	LHC Commissioning & Upgrade	M. Giovannozzi	78443
ABP-SU	Survey	D. Missiaen	75483

CO - Controls

E. Hatziangeli	Group Leader		75239
M. Draper	Deputy Group Leader		73348
E. Gavrill	*Secretariat*	*Fax : 7675800*	*74788*
CO-DA	Data and Applications	K. Sigerud	71464
CO-DO	DevOps	V. Baggiolini	73401
CO-FE	Front-Ends	M. Vanden Eynden	75578
CO-HT	Hardware & Timing	J. Serrano	73486
CO-IN	Infrastructure	P. Charrue	75410

ASR - Administration, Safety & Resources

R. Billen	Group Leader		76904
M. Tavlet	Deputy Group Leader		73717
F. Berrenger	*Secretariat*	*Fax : 7667935*	*75126*
ASR-AS	Administration & Secretariats	L. Van Cauter Tanner	78575
ASR-RL	Resources & Logistics	R. Billen	76904
ASR-SU	Safety Unit	M. Tavlet	73717

OP - Operation

M. Lamont	Group Leader		74806
M. Benedikt	Deputy Group Leader		73380
S. Dubourg	*Secretariat*	*Fax : 7668296*	*73631*
OP-AD	Antiproton Decelerator	T. Eriksson	74568
OP-LHC	Large Hadron Collider	J. Wenninger	75379
OP-MED	MedAustron	M. Benedikt	73380
OP-PS	Proton Synchrotron	R. Steerenberg	79086
OP-PSB	Proton Syn. Booster & Isolde op.	K. Hanke	73575
OP-SPS	Super Proton Synchrotron	K. Cornelis	75209
OP-TI	Technical Infrastructure	P. Sollander	78081

BI - Beam Instrumentation

R. Jones	Group Leader		74783
J.-J. Gras	Deputy Group Leader		79577
M. Catin	*Secretariat*	*Fax : 7677740*	*74619*
BI-BL	Beam Loss Monitoring	B. Dehning	75541
BI-EA	Experimental Areas	G. Tranquille	70243
BI-QP	Tune & Position	T. Lefevre	74149
BI-ML	Mechanics & Logistics	R. Veness	73009
BI-PI	Position & Intensity	L. Soby	72632
BI-PM	Profile Measurements	E. Bravin	71885
BI-SW	Software Support	L. Jensen	71772

RF - Radio Frequency

E. Jensen	Group Leader		74298
M. Vretenar	Deputy Group Leader		72925
L. Hemery	*Secretariat*	*Fax : 7678509*	*62142*
RF-BR	Beams & RF Studies	E. Chapochnikova	76778
RF-CS	Servos & Controls Interface	A. Butterworth	79806
RF-FB	Feedbacks & Beam Control	W. Hofle	75175
RF-IS	Injector Synchrotrons	C. Rossi	74563
RF-MK	Modulators & Klystrons	O. Brunner	76251
RF-LRF	Linacs RF	F. Gerigk	70562
RF-PM	Power Amplifiers, Couplers & Modules Production	E. Montesinos	75190
RF-SRF	SC RF Cavity Technology	K. Schirm	76770

BE/ASR/AS/lvc

12/03/2013

Annexe 3
Structure du département ABP

ABP GROUP STRUCTURE

Functions

LMC: LHC Machine Coordinator

MSI: Machine Superintendent

(1): LINAC4 Dep. Project Leader

(3): MADX Custodian

(5): MTE Beam Commissioning Coordinator, PS- LIU project coordinator

(7): MTE Project Leader

(9): LHC Background WG Coordinator

(11) HL-LHC Deputy Project Leader

(13): EuCard Coordinator

(15): ELENA Deputy Project Leader

(17): HL-LHC WP2 Coordinator

(19): HL-LHC WP5 Coordinator

MS: Machine Supervisor

T : TSO

(2): PS-SPS Complex Machine Development Coordinator

(4) EuroNu Beta-Beams WP Leader

(6): ELENA Project Leader

(8): Collimation Project Leader

(10): Computer Administration

(12): Ion Machine coordinator

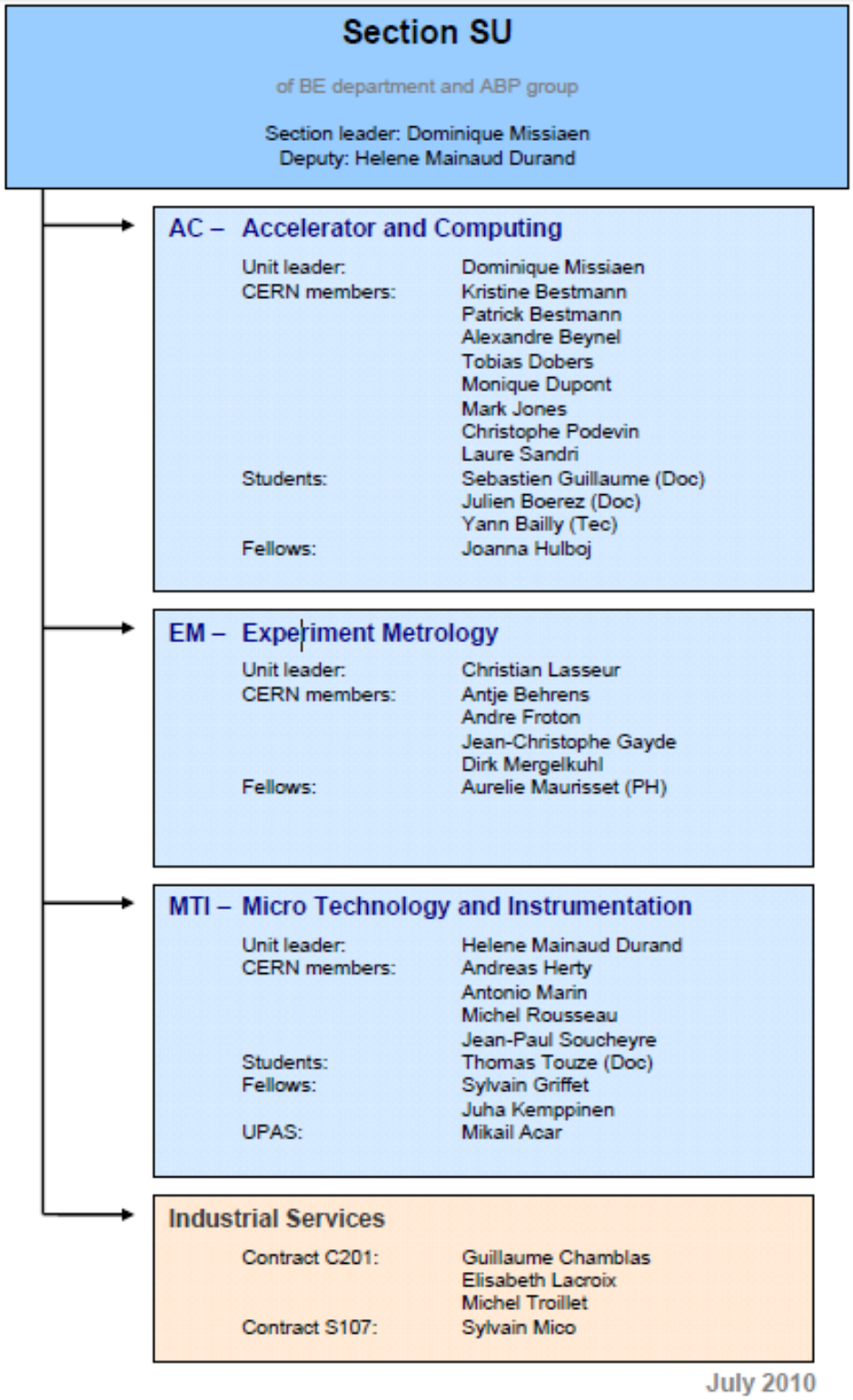
(14): CLIC-CTF3 Collaboration Spoke-person

(16) : SixTrack Custodian

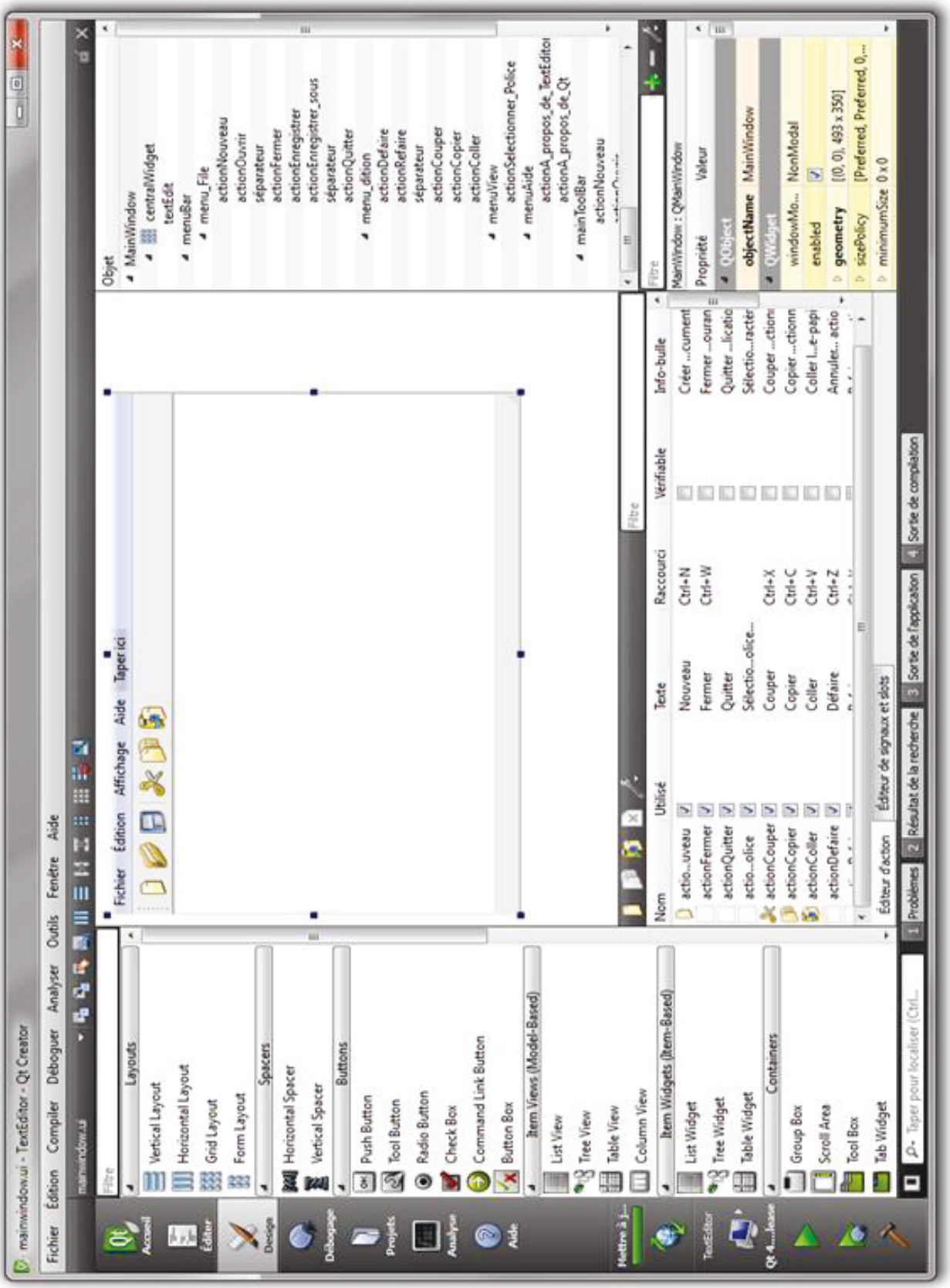
(18) : HL-LHC WP8 Coordinator

** Part Time*

Annexe 4
Structure de la section SU



Annexe 5
Qt Designer – Ecran d'édition des fenêtres



Annexe 6
Points de fonction – Dénombrement des composants

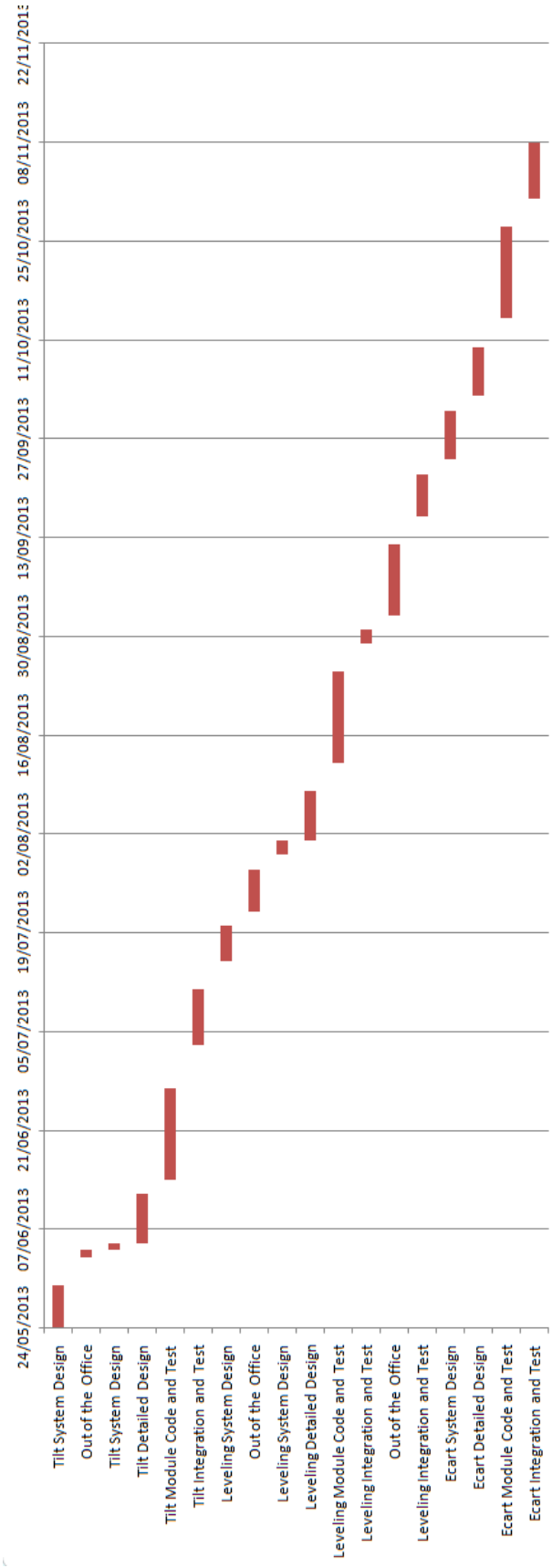
Activité	Fonction	Points de fonction						Données élémentaires (DE)	Sous-ensemble logique de données (SLD)	Groupe de données référencées
		oupe de données inter	oupes de données extern	Entrées	Sorties	terrogation	Complexité (Faible, Moyenne,			
Selection de la tâche	Menu de démarrage			1			F	1		1
Navigation entre les fenêtres	Retour vers menu démarrage			1			F	1		1
	Quitter application			1			F	1		1
Nivellement – Réglage des références	Choix de l'équipe			1			F	1		1
	Choix de l'instrument			1			F	3		1
	Choix de la mire			1			F	3		1
	Selection du fichier Théo			1			F	16		1
	Chargement du fichier Théo	1					F	16	1	1
	Sélection du fichier Séqu			1			F	4		1
	Chargement du fichier Séqu	1					F	4	1	1
	Choix de l'opération			1			F	3		1
Nivellement – Selections des points à mesurer	Afficher la liste des points									
	Choix Aller - Retour			1			F	1		1
	Sélections des points manuellements			1			F	4		2
	Ajout d'un point manuellement			1			F	4		1
	Recherche d'un point					1	F	4		1
	Afficher la séquence de points									
	Sélection d'une séquence de points			1			F	4		2
	Suppression d'un point			1			F	4		1
	Réordonner la liste de points			1			F	4		1
	Trier la liste de points			1			F	4		1
Nivellement – Enregistrement des mesures	Afficher la liste des points									
	Afficher la liste des points suivante (nouvelle station)									
	Ecriture des mesures			1			F	4	2	2
	Calcul de l'écart (niv. Retour)	1					F	8		
	Calcul de l'emq (niv. Retour)	1					F	8	2	
	Affichage du message 'reprise de la station' (emq mauvais									
	Création du fichier retour				1		F	6		1
	Enregistrement des mesures			1			F	1		1
	Enregistrement des mesures				1		E	23		1
	Affichage des mesures précédentes									
Ecartométrie – Réglage des	Masquer le menu 'mire'									
Ecartométrie – Selections des points à mesurer	Afficher la liste des points									
	Sélections des points manuellements			1			F	4		2
	Ajout d'un point manuellement			1			F	4		1
	Recherche d'un point									
	Afficher la séquence de points									
	Sélection d'une séquence de points			1			F	4		2
	Suppression d'un point									
	Réordonner la liste de points									
	Trier la liste de points									
	Insertion d'un pilier entre les alésages sélectionnés			1			F	4		1
	Affichage des points d'ancrage (Séqu)									
	Sélection des points d'ancrage (Séqu)			1			F	2		1

		Points de fonction								
Activité	**Fonction**	**oupe de données intern**	**oupes de données extern**	**Entrées**	**Sorties**	**nterrogation**	**Complexité (Faible, Moyenne,**	**Données élémentaires (DE)**	**Sous-ensemble logique de données (SLD)**	**Groupe de données référencées**
Ecartométrie – Selections des points à mesurer	Afficher la liste des points									
	Sélections des points manuellements			1			F	4		2
	Ajout d'un point manuellement			1			F	4		1
	Recherche d'un point									
	Afficher la séquence de points									
	Sélection d'une séquence de points			1			F	4		2
	Suppression d'un point									
	Réordonner la liste de points									
	Trier la liste de points									
	Insertion d'un pilier entre les alésages sélectionnés			1			F	4		1
	Affichage des points d'ancrage (Sequ)									
	Sélection des points d'ancrage (Sequ)			1			F	2		1
Ecartométrie – Enregistrement des mesures	Afficher la liste des points									
	Afficher la liste des points prédèdent (fil précèdent)									
	Afficher la liste des points suivante (fil suivant)									
	Ecriture des mesures			1			F	3		2
	Calcul de l'écart	1					F	6	1	
	Enregistrement des mesures			1			F	1		1
	Enregistrement des mesures				1		E	22		1
	Affichage des mesures précèdentes					1	F	3		1
Tilt – Réglage des références	Masquer le menu 'mire'									
	Masquer le menu 'opération'									
	Choix de l'opération init.			1			F	3		1
	Choix de l'opération régl.			1			F	3		1
Tilt – Selections des points à mesurer	Afficher la liste des points									
	Sélections des points manuellements			1			F	4		2
	Ajout d'un point manuellement			1			F	4		1
	Recherche d'un point									
	Suppression d'un point									
	Réordonner la liste de points									
	Trier la liste de points									
Tilt – Enregistrement des mesures	Afficher la liste des points									
	Afficher le point précèdent (mes. Init.)									
	Afficher le point courant (mes. Init.)									
	Afficher le point suivant (mes. Init.)									
	Afficher le point précèdent (mes. Préc.)									
	Afficher le point courant (mes. Préc.)									
	Afficher le point suivant (mes. Préc.)									
	Afficher le point précèdent (mes. Régl.)									
	Afficher le point courant (mes. Régl.)									
	Afficher le point suivant (mes. Régl.)									
	Ecriture des mesures			1			F	4		2
	Calcul de la lecture moyenne	1					F	2	1	
	Affichage de la lecture théorique									
	Calcul de l'écart (mrad)	1					F	2	1	
	Enregistrement des mesures			1			F	1		1
	Enregistrement des mesures				1		E	23		1

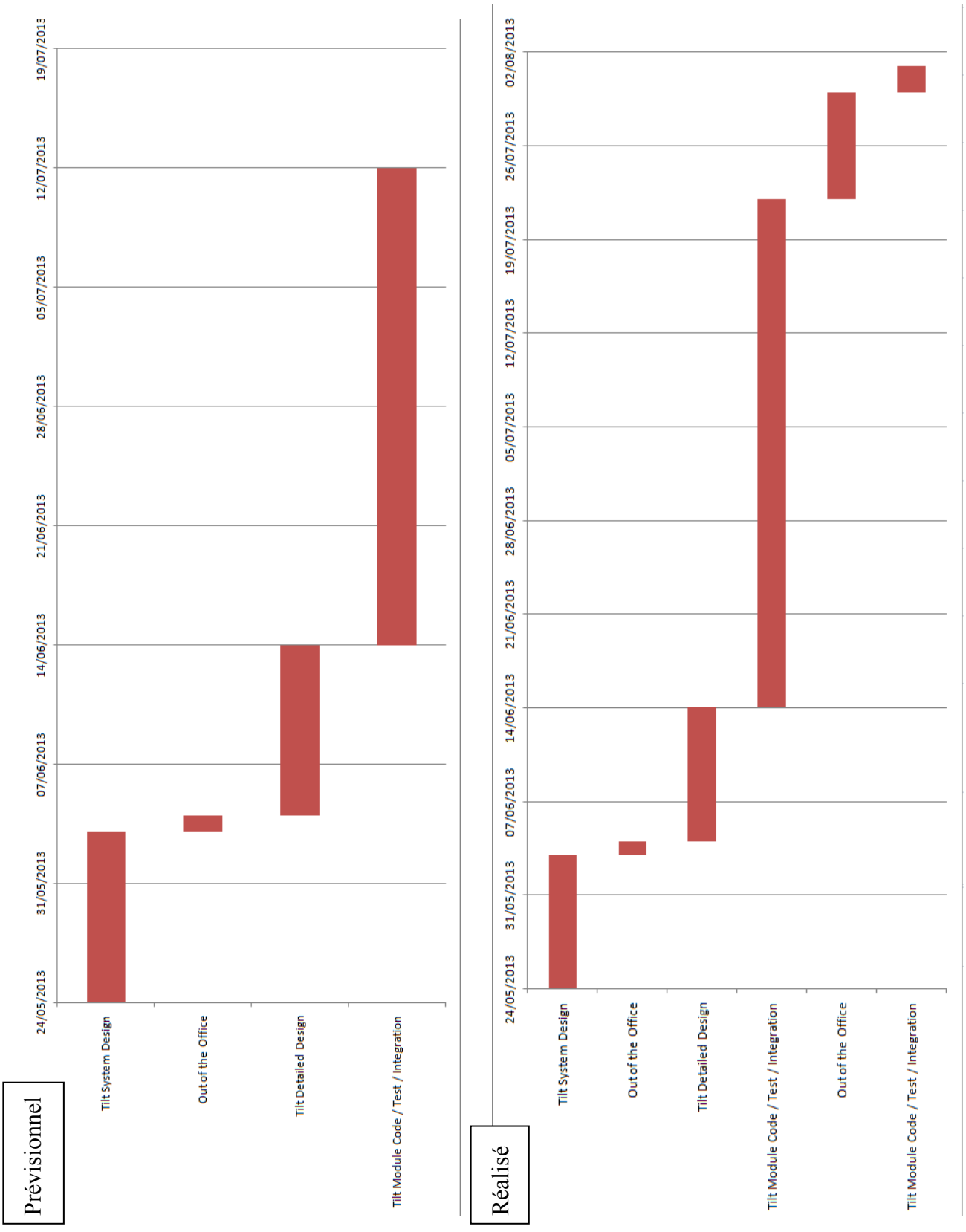
Annexe 7
Points de fonction – Addition des points de fonction

Type de composant	Complexité	Nombre		Total par complexité		Totaux par types de composant
GDI	F	7	x	7	=	49
	M	0	x	10	=	0
	E	0	x	15	=	0
GDE	F	0	x	5	=	0
	M	0	x	7	=	0
	E	0	x	10	=	0
ENT	F	31	x	3	=	93
	M	0	x	4	=	0
	E	0	x	6	=	0
SOR	F	1	x	4	=	4
	M	0	x	5	=	0
	E	3	x	7	=	21
INT	F	4	x	3	=	12
	M	0	x	4	=	0
	E	0	x	6	=	0
		Nombre de points de fonction bruts				**179**

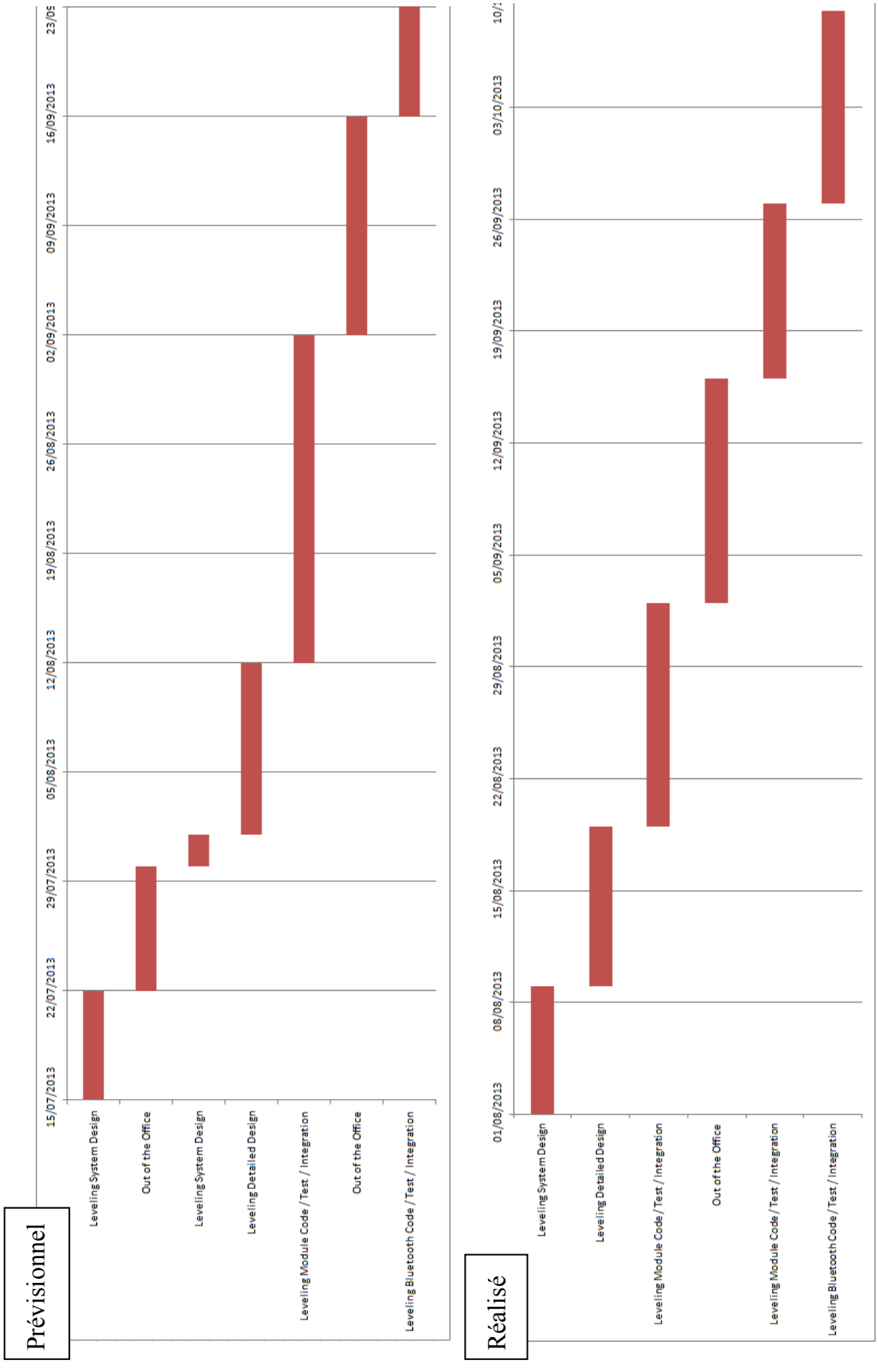
Annexe 8
Planning prévisionnel de développement – Diagramme de Gantt



Annexe 9
Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Tilt

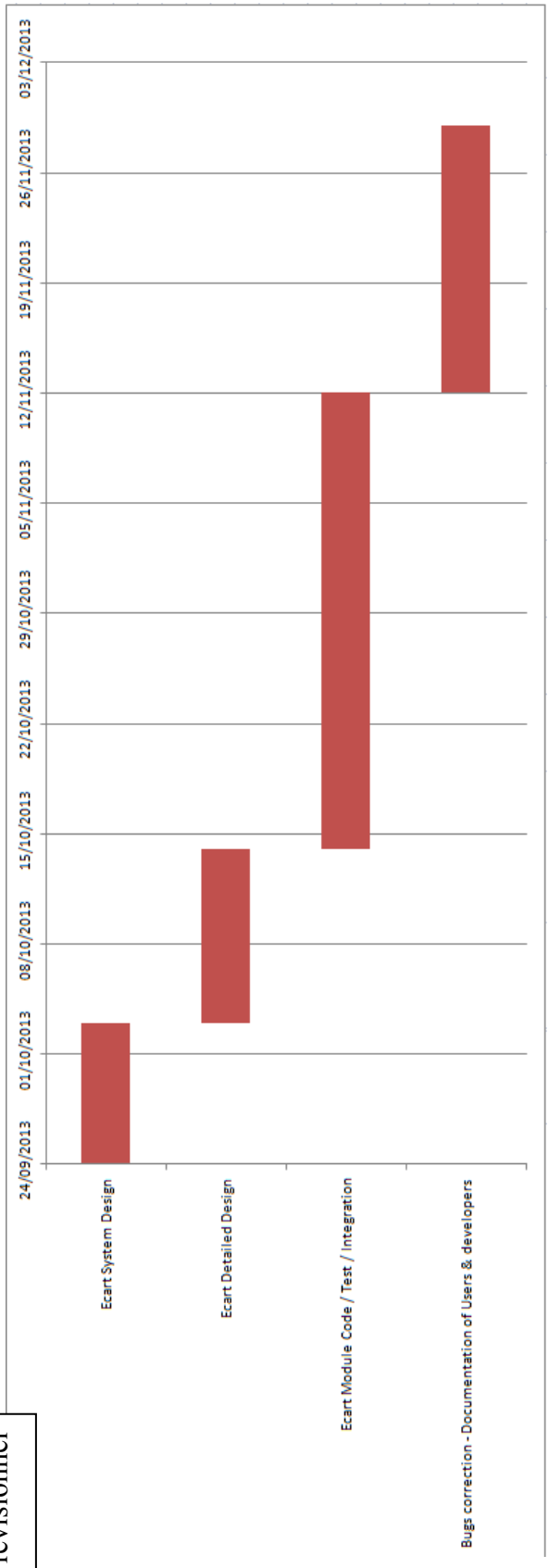


Annexe 10
Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Nivellement

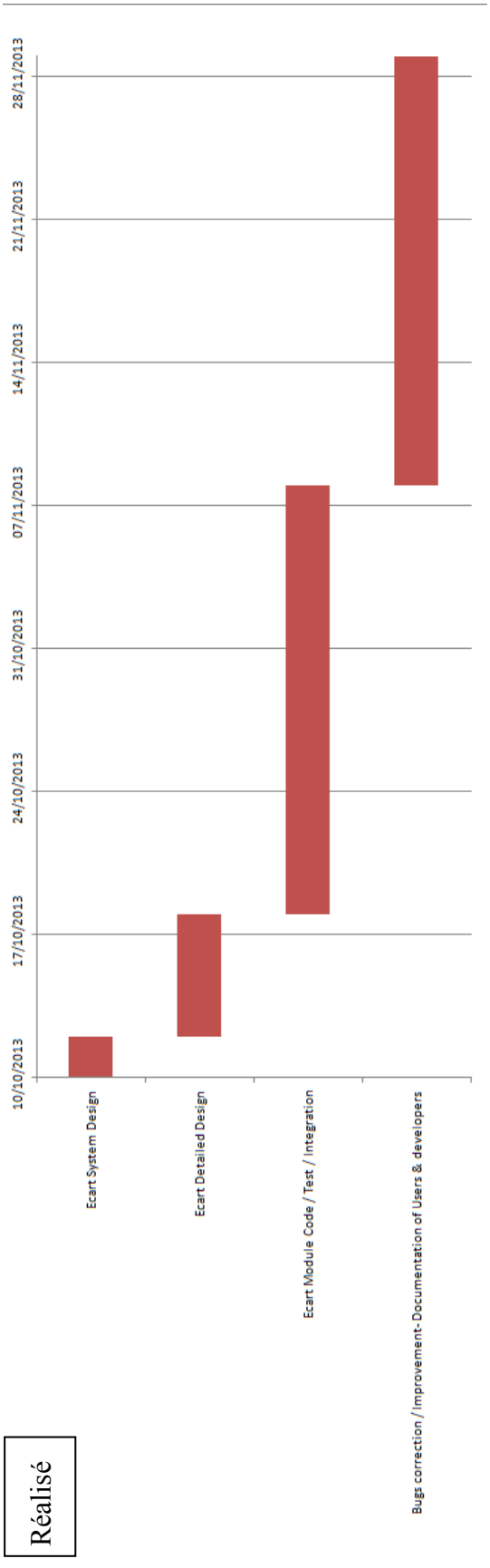


Annexe 11
Comparaison planning prévisionnel et réalisé – Diagramme de Gantt – Module Ecartométrie

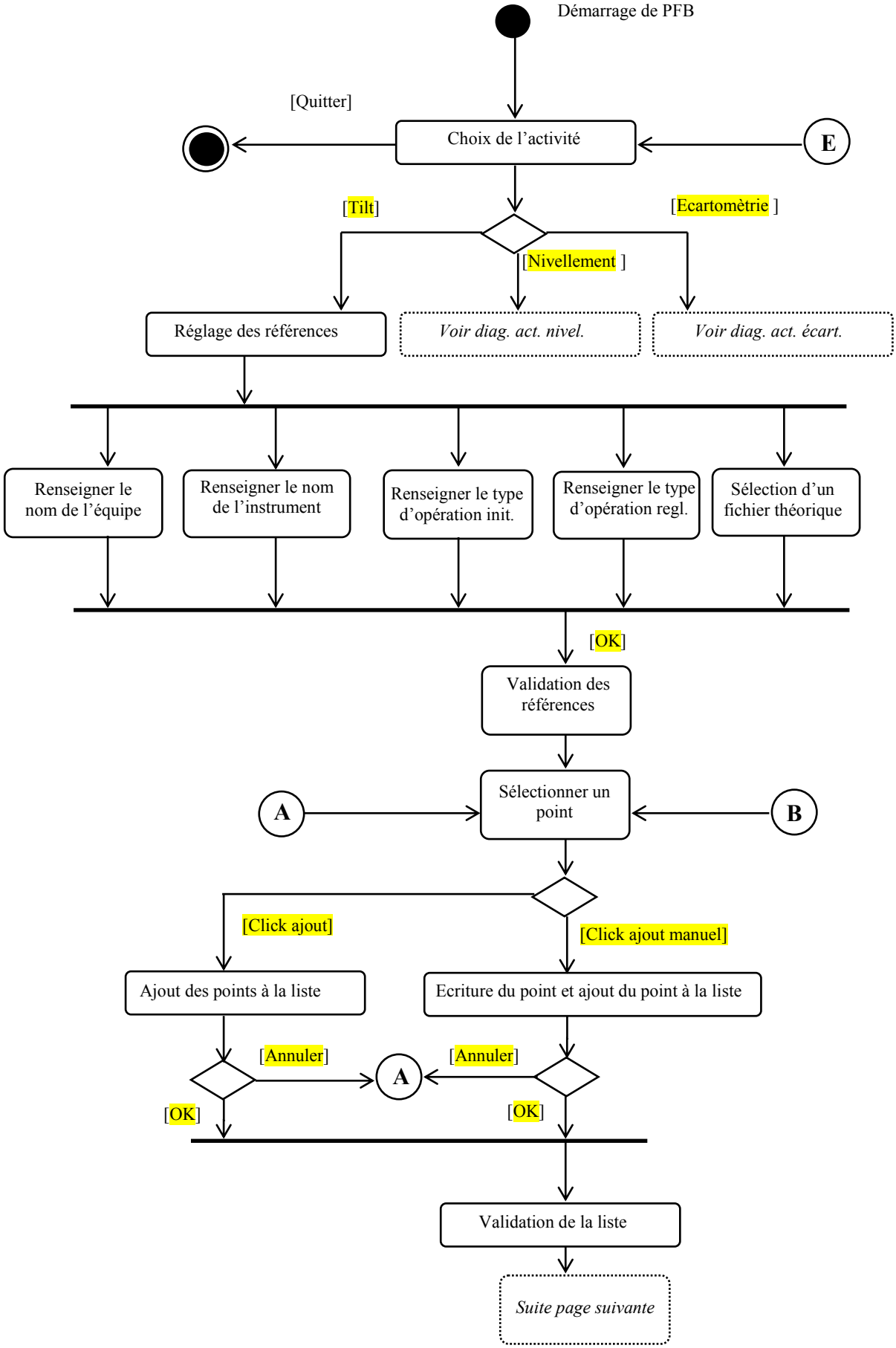
Prévisionnel

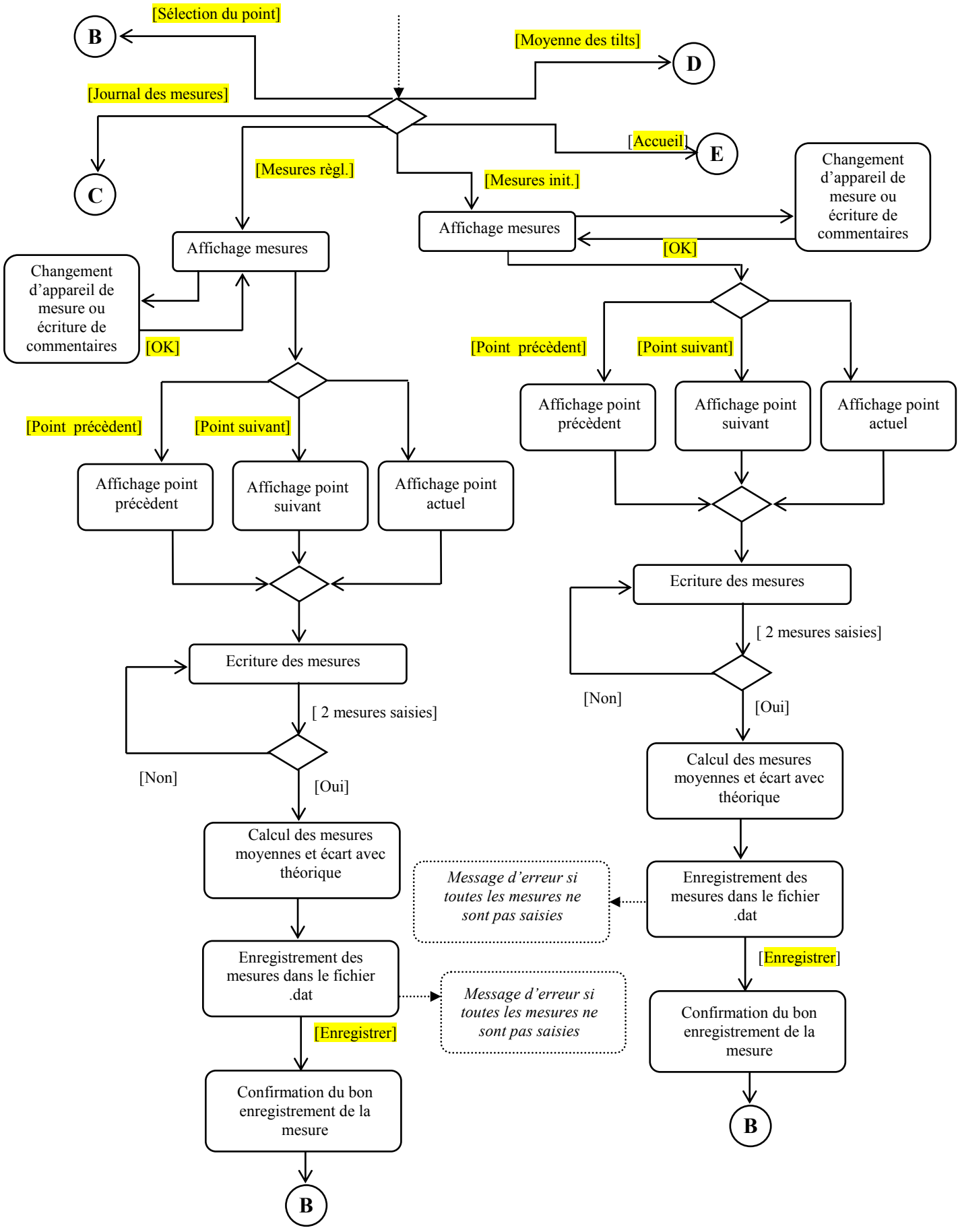


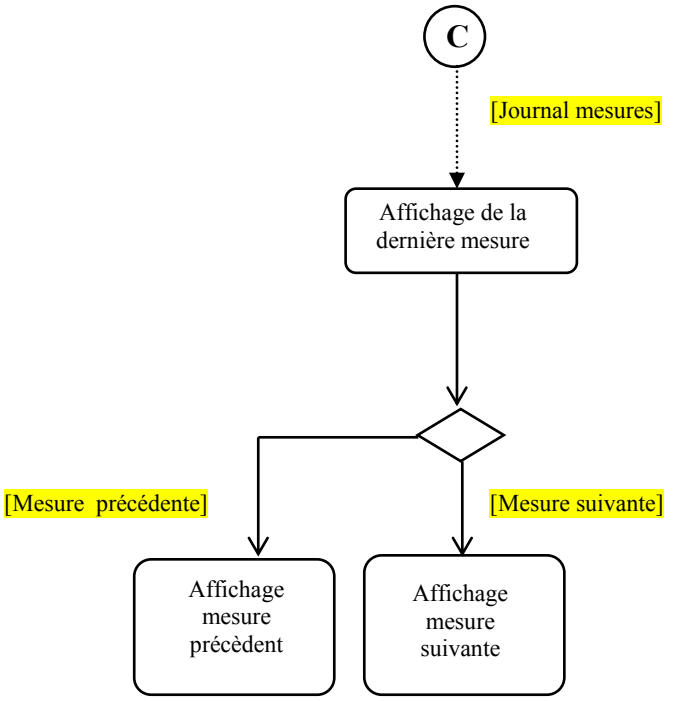
Réalisé

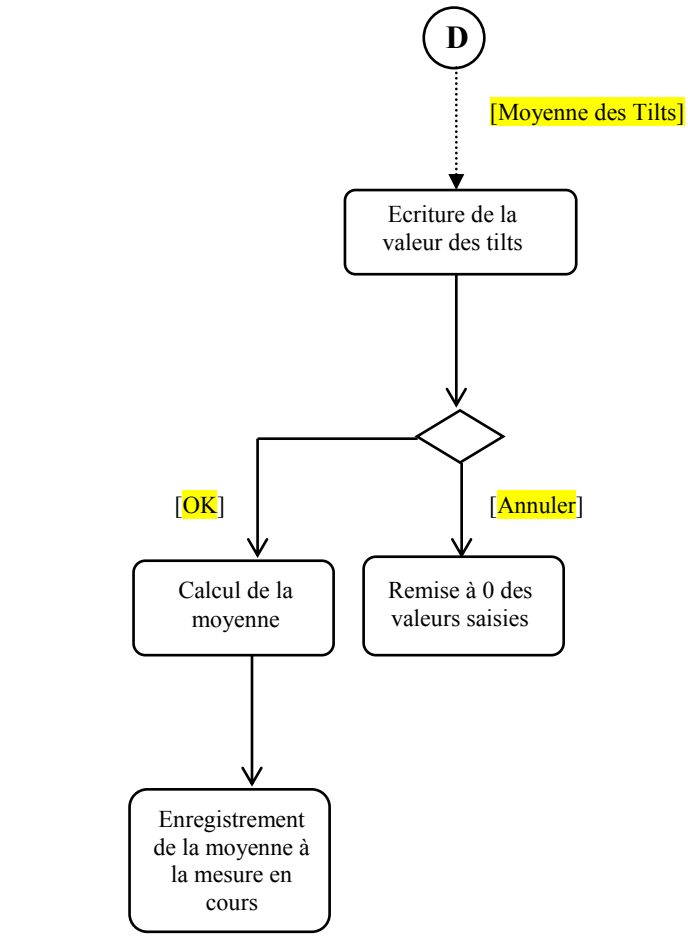


Annexe 12
Diagramme d'activité – Module de tilt

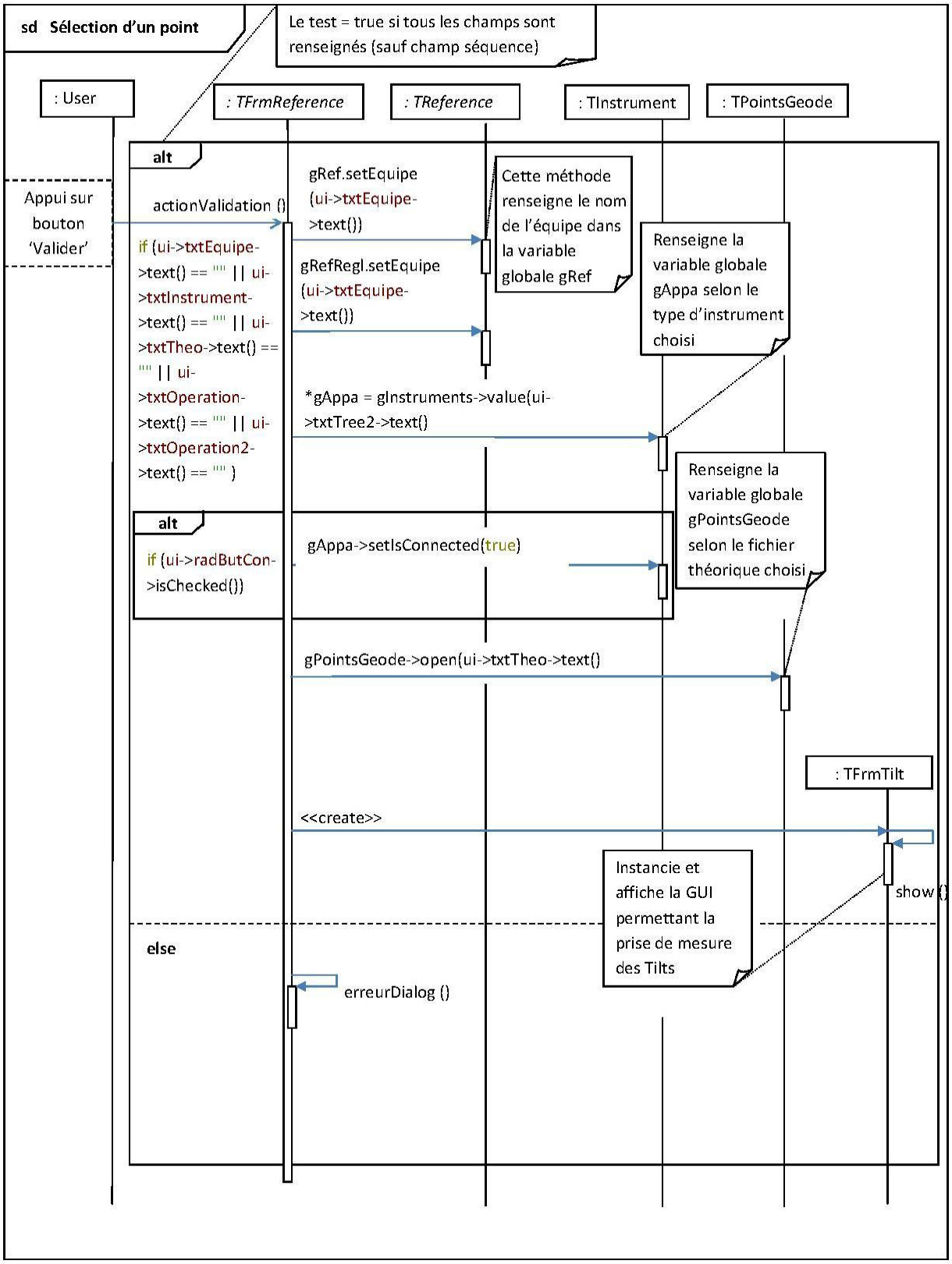



B
[Sélection du point]
[Moyenne des tilts]
D
[Journal des mesures]
C
[Accueil]
E
[Mesures règl.]
[Mesures init.]
Changement d'appareil de mesure ou écriture de commentaires
Affichage mesures
[OK]
Changement d'appareil de mesure ou écriture de commentaires
[OK]
[Point précèdent]
[Point suivant]
Affichage point précèdent
Affichage point suivant
Affichage point actuel
[Point précèdent]
[Point suivant]
Affichage point précèdent
Affichage point suivant
Affichage point actuel
Ecriture des mesures
[2 mesures saisies]
[Non]
[Oui]
Calcul des mesures moyennes et écart avec théorique
Message d'erreur si toutes les mesures ne sont pas saisies
Enregistrement des mesures dans le fichier .dat
[Enregistrer]
Confirmation du bon enregistrement de la mesure
B
Ecriture des mesures
[2 mesures saisies]
[Non]
[Oui]
Calcul des mesures moyennes et écart avec théorique
Enregistrement des mesures dans le fichier .dat
Message d'erreur si toutes les mesures ne sont pas saisies
[Enregistrer]
Confirmation du bon enregistrement de la mesure
B

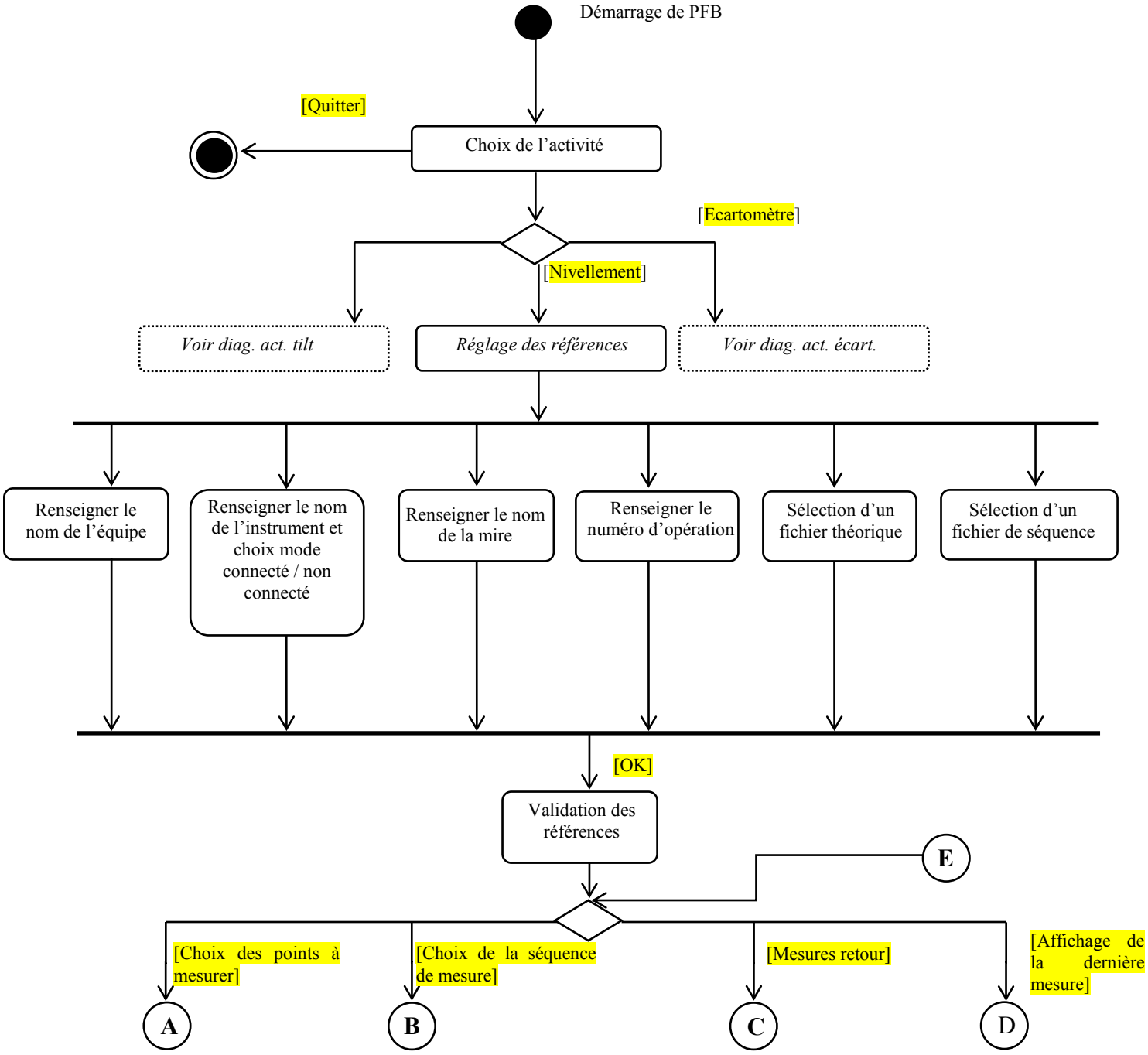

C
[Journal mesures]
Affichage de la dernière mesure
[Mesure précédente]
[Mesure suivante]
Affichage mesure précèdent
Affichage mesure suivante

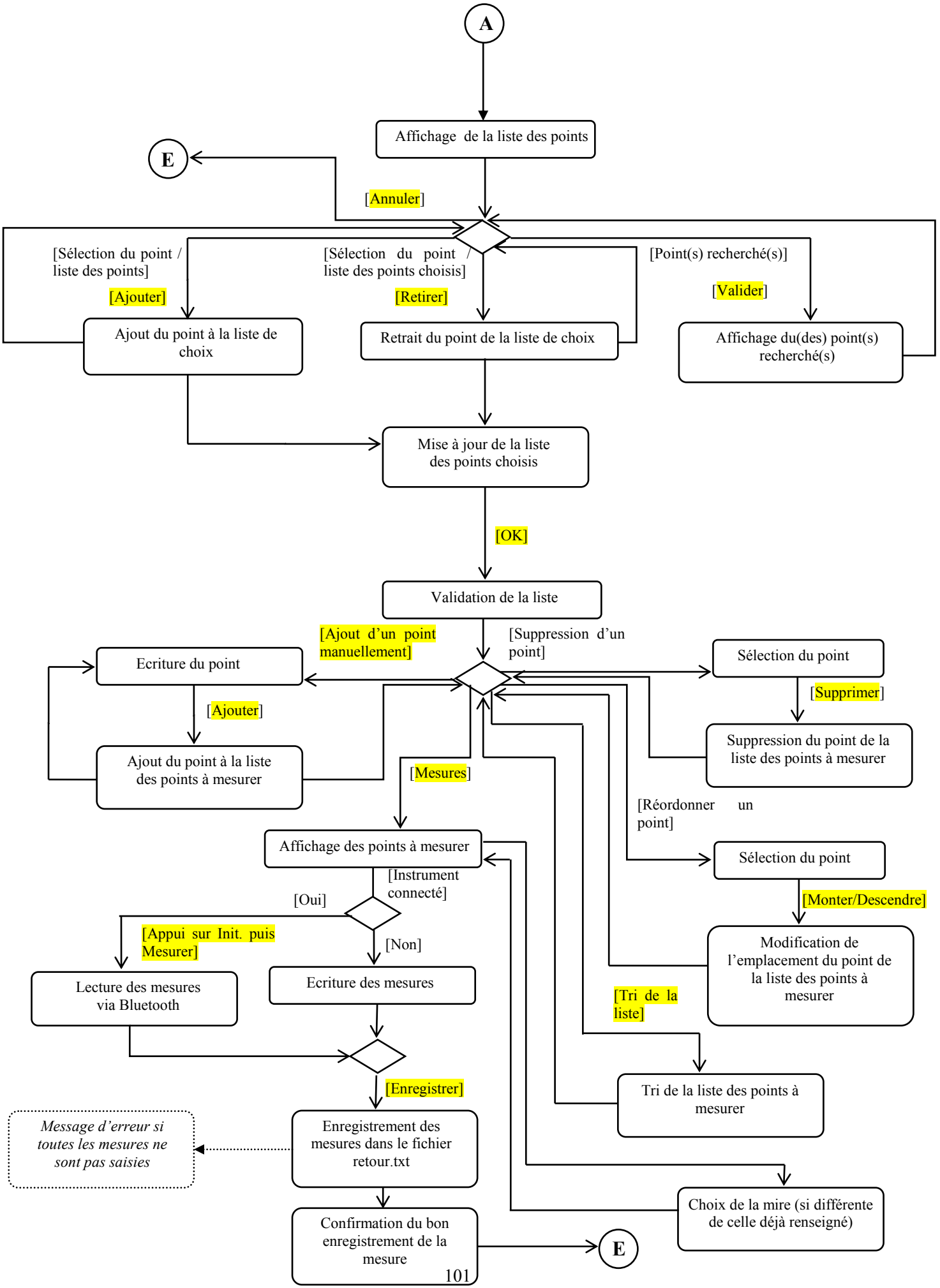

D
[Moyenne des Tilts]
Ecriture de la valeur des tilts
[OK]
[Annuler]
Calcul de la moyenne
Remise à 0 des valeurs saisies
Enregistrement de la moyenne à la mesure en cours

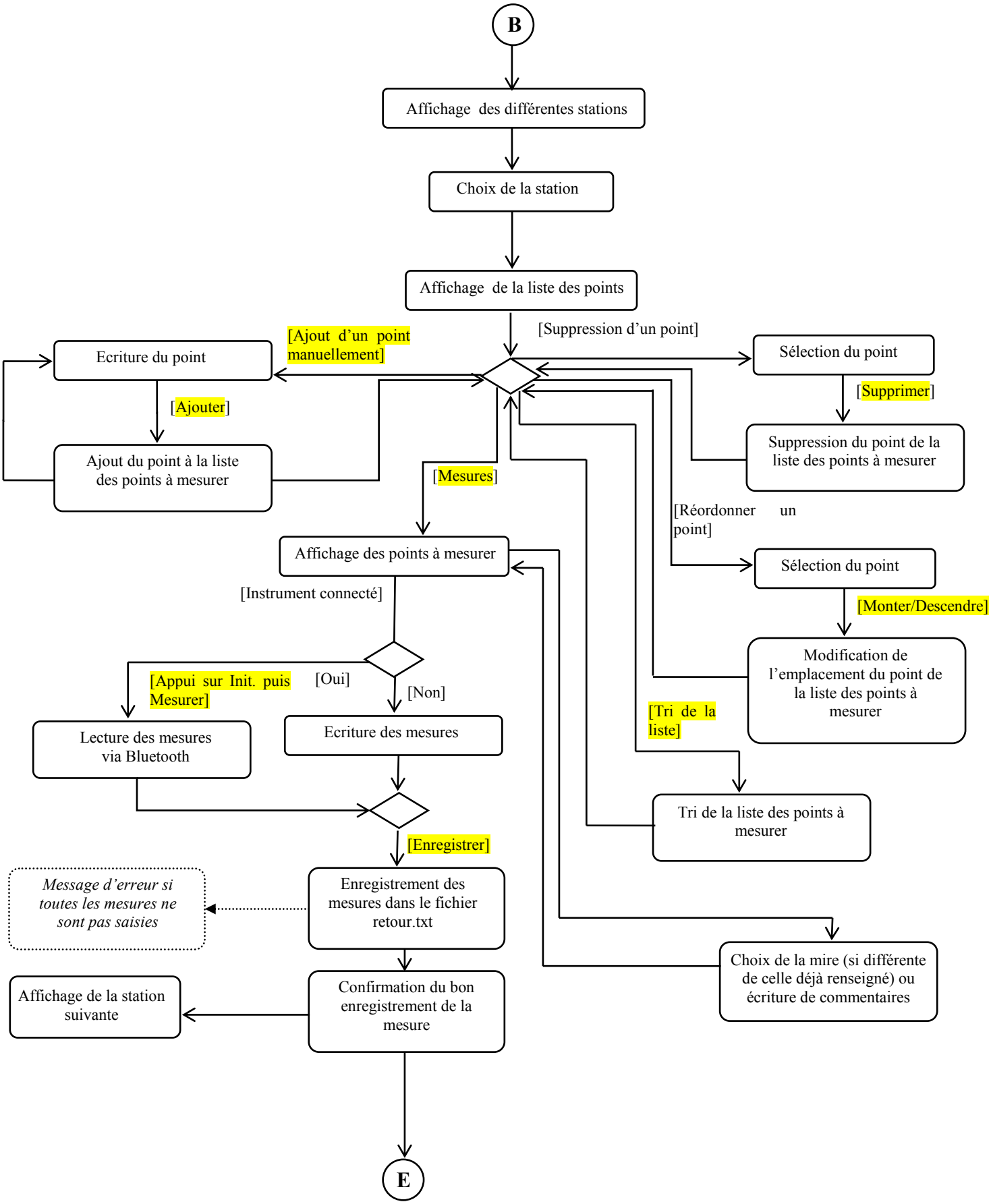
Annexe 13
Diagramme de séquence – Module de tilt – Validation des références

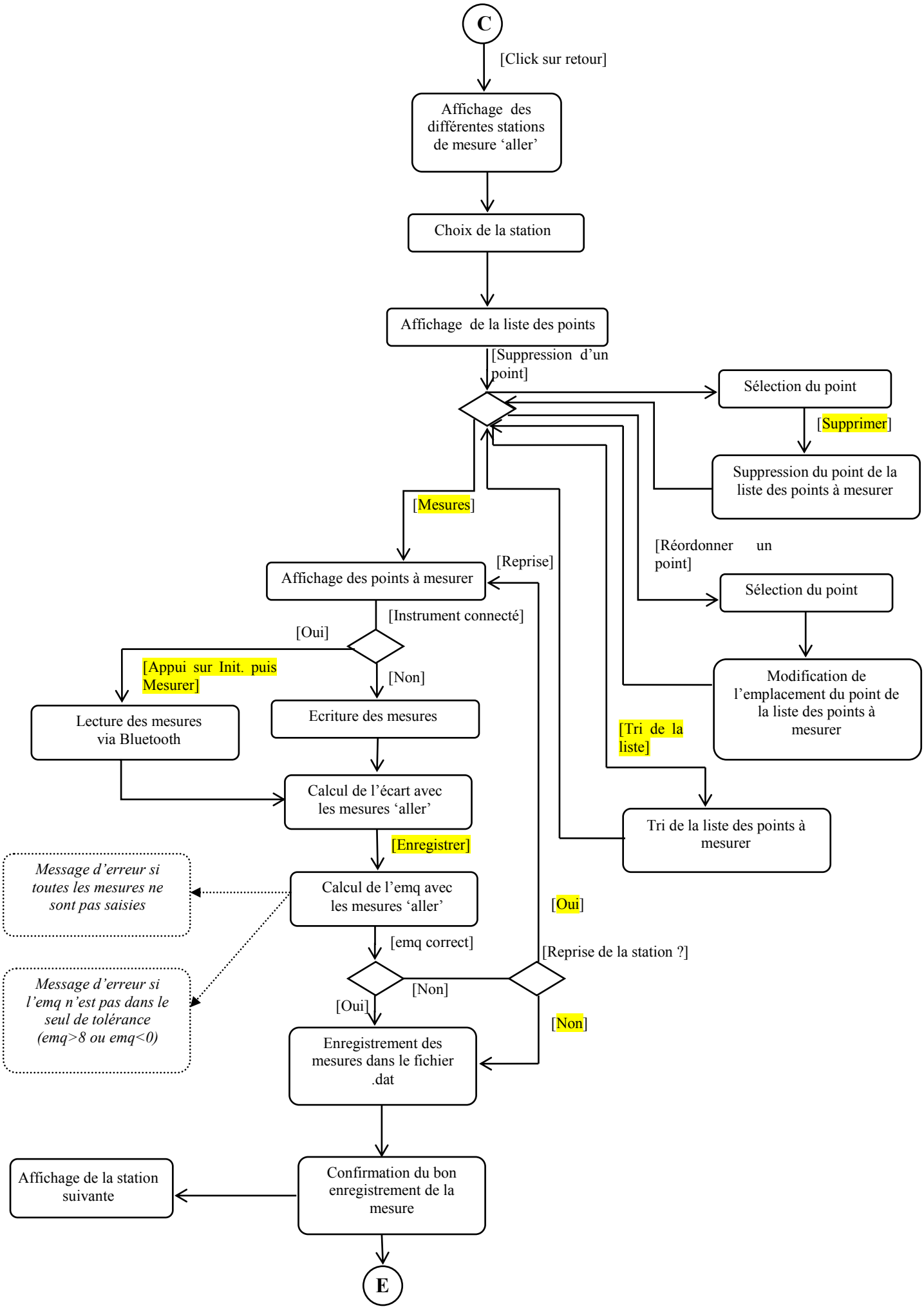


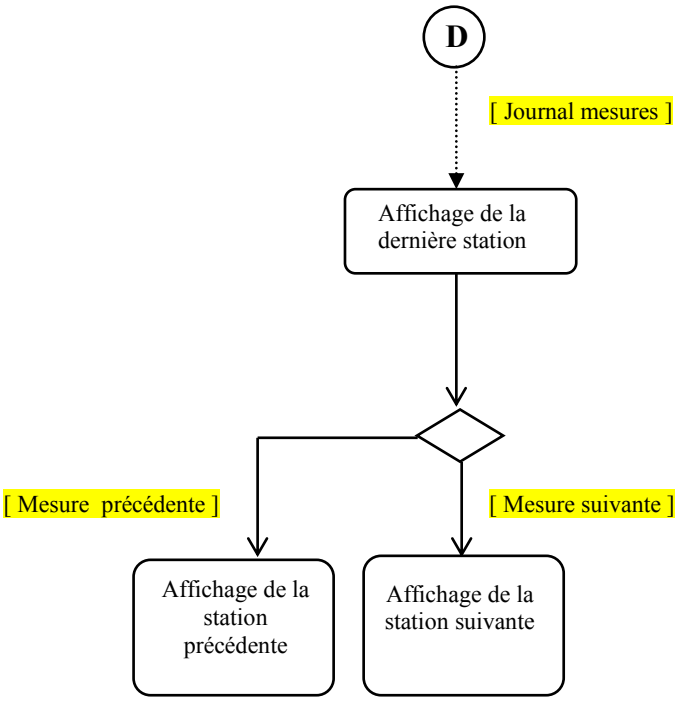
Annexe 14
Diagramme d'activité – Module de nivellement



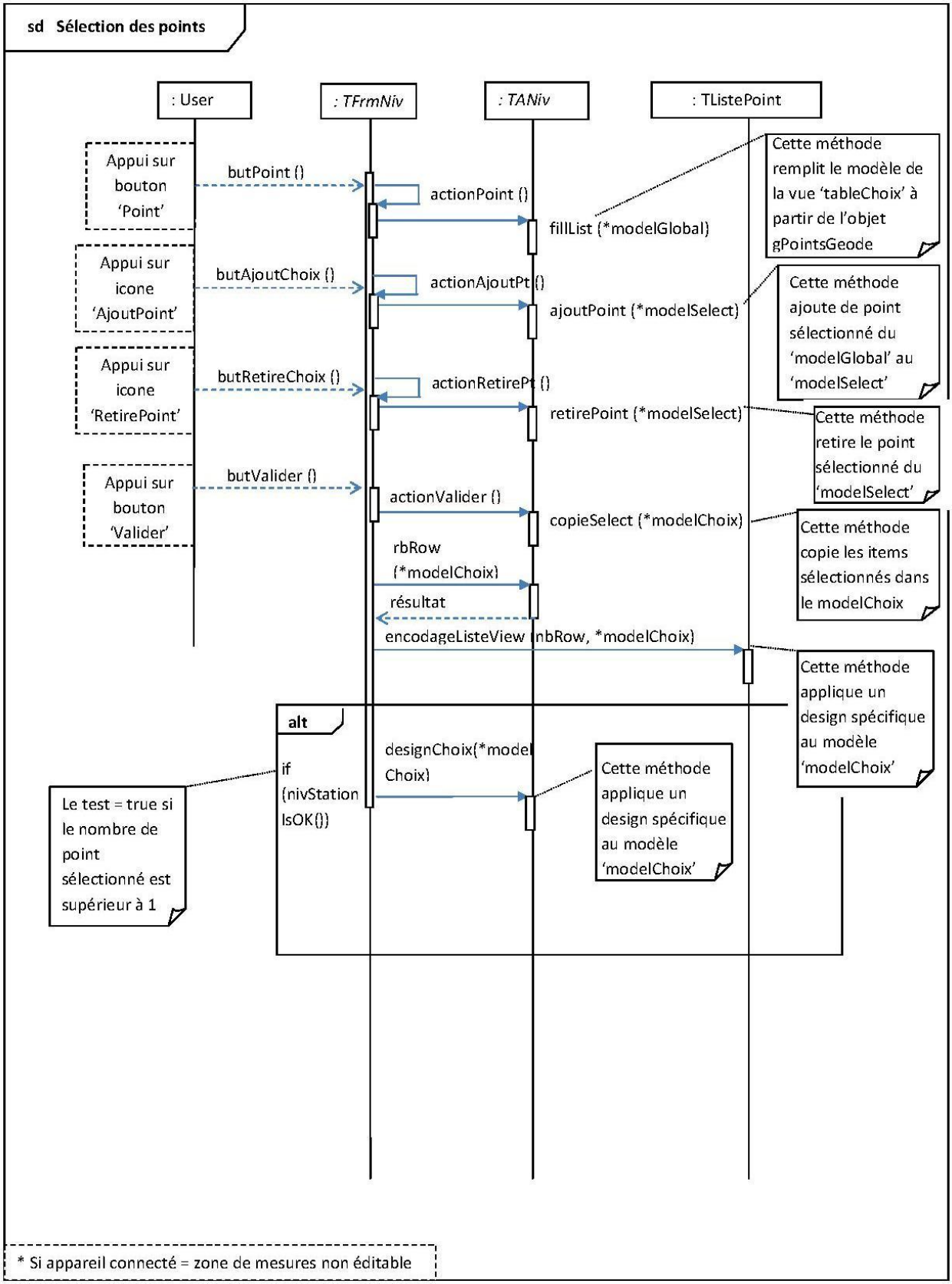

A
Affichage de la liste des points
E
[Annuler]
[Sélection du point / liste des points]
[Ajouter]
Ajout du point à la liste de choix
[Sélection du point / liste des points choisis]
[Retirer]
Retrait du point de la liste de choix
[Point(s) recherché(s)]
[Valider]
Affichage du(des) point(s) recherché(s)
Mise à jour de la liste des points choisis
[OK]
Validation de la liste
[Ajout d'un point manuellement]
[Suppression d'un point]
Ecriture du point
[Ajouter]
Ajout du point à la liste des points à mesurer
Sélection du point
[Supprimer]
Suppression du point de la liste des points à mesurer
[Mesures]
[Réordonner un point]
Affichage des points à mesurer
Sélection du point
[Monter/Descendre]
Modification de l'emplacement du point de la liste des points à mesurer
[Instrument connecté]
[Oui]
[Non]
[Appui sur Init. puis Mesurer]
Lecture des mesures via Bluetooth
Ecriture des mesures
[Tri de la liste]
Tri de la liste des points à mesurer
[Enregistrer]
Message d'erreur si toutes les mesures ne sont pas saisies
Enregistrement des mesures dans le fichier retour.txt
Choix de la mire (si différente de celle déjà renseigné)
Confirmation du bon enregistrement de la mesure
E


B
Affichage des différentes stations
Choix de la station
Affichage de la liste des points
[Suppression d'un point]
[Ajout d'un point manuellement]
Ecriture du point
[Ajouter]
Ajout du point à la liste des points à mesurer
Sélection du point
[Supprimer]
Suppression du point de la liste des points à mesurer
[Mesures]
[Réordonner un point]
Affichage des points à mesurer
Sélection du point
[Monter/Descendre]
[Instrument connecté]
[Appui sur Init. puis Mesurer]
[Oui]
[Non]
Modification de l'emplacement du point de la liste des points à mesurer
[Tri de la liste]
Lecture des mesures via Bluetooth
Ecriture des mesures
Tri de la liste des points à mesurer
[Enregistrer]
Message d'erreur si toutes les mesures ne sont pas saisies
Enregistrement des mesures dans le fichier retour.txt
Choix de la mire (si différente de celle déjà renseigné) ou écriture de commentaires
Affichage de la station suivante
Confirmation du bon enregistrement de la mesure
E

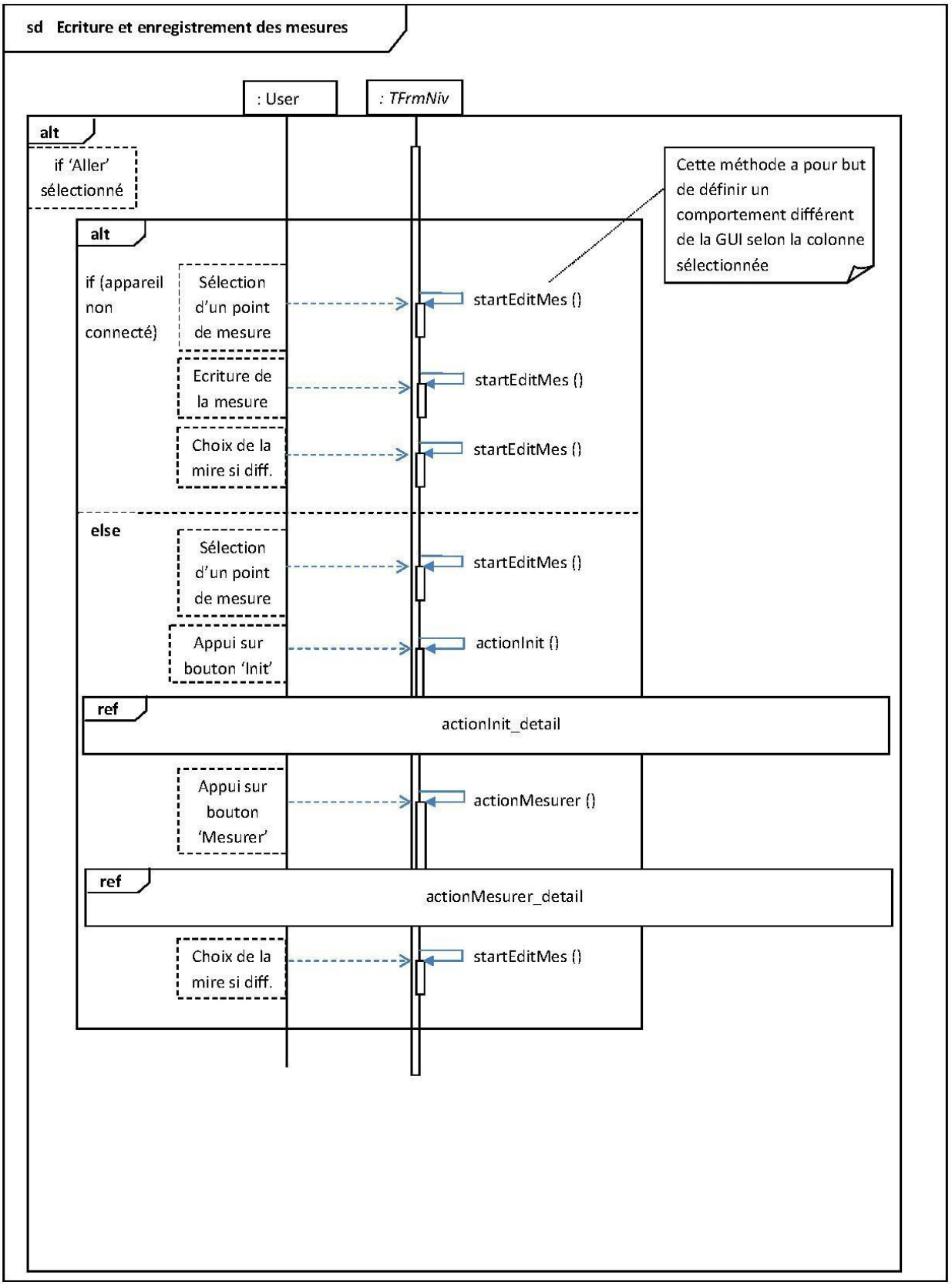

C
[Click sur retour]
Affichage des différentes stations de mesure 'aller'
Choix de la station
Affichage de la liste des points
[Suppression d'un point]
Sélection du point
[Supprimer]
Suppression du point de la liste des points à mesurer
[Mesures]
[Réordonner un point]
Sélection du point
Modification de l'emplacement du point de la liste des points à mesurer
[Tri de la liste]
Tri de la liste des points à mesurer
[Reprise]
Affichage des points à mesurer
[Instrument connecté]
[Oui]
[Appui sur Init. puis Mesurer]
[Non]
Lecture des mesures via Bluetooth
Ecriture des mesures
Calcul de l'écart avec les mesures 'aller'
[Enregistrer]
Message d'erreur si toutes les mesures ne sont pas saisies
Calcul de l'emq avec les mesures 'aller'
[Oui]
[emq correct]
[Reprise de la station ?]
Message d'erreur si l'emq n'est pas dans le seul de tolérance (emq>8 ou emq<0)
[Non]
[Oui]
[Non]
Enregistrement des mesures dans le fichier .dat
Affichage de la station suivante
Confirmation du bon enregistrement de la mesure
E

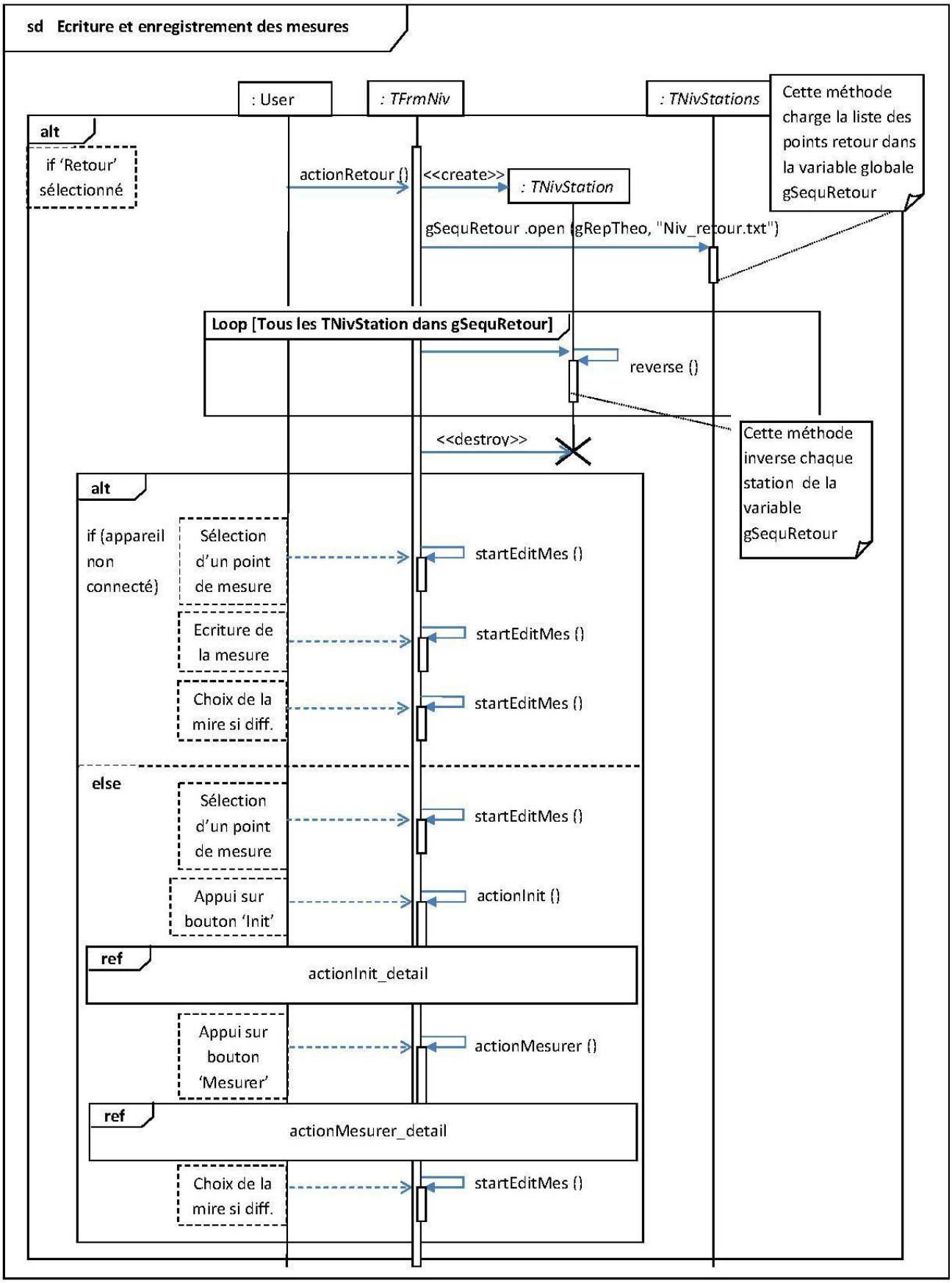

D
[Journal mesures]
Affichage de la dernière station
[Mesure précédente]
[Mesure suivante]
Affichage de la station précédente
Affichage de la station suivante

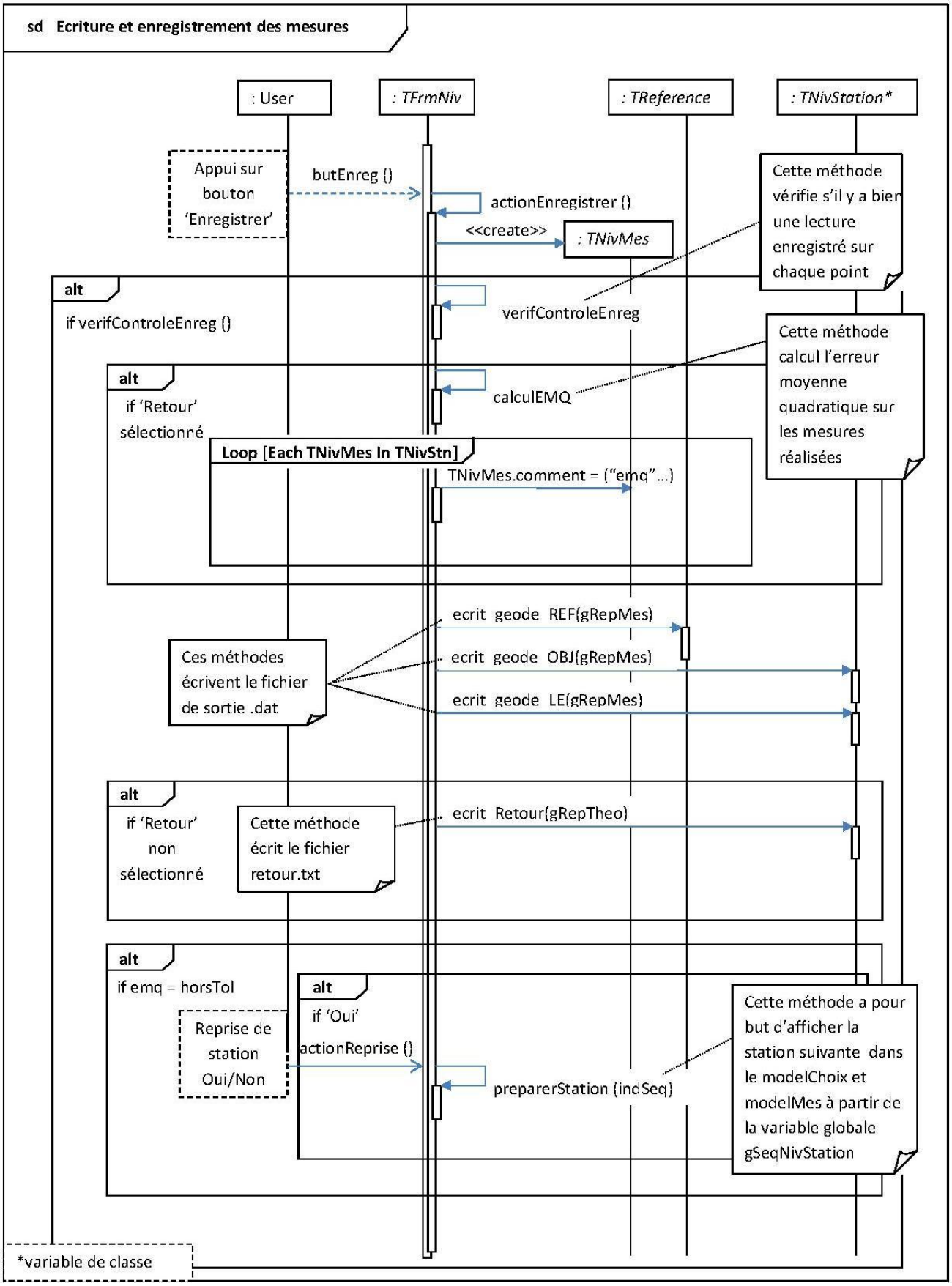
Annexe 15
Diagramme de séquence – Module de nivellement – Sélection des points

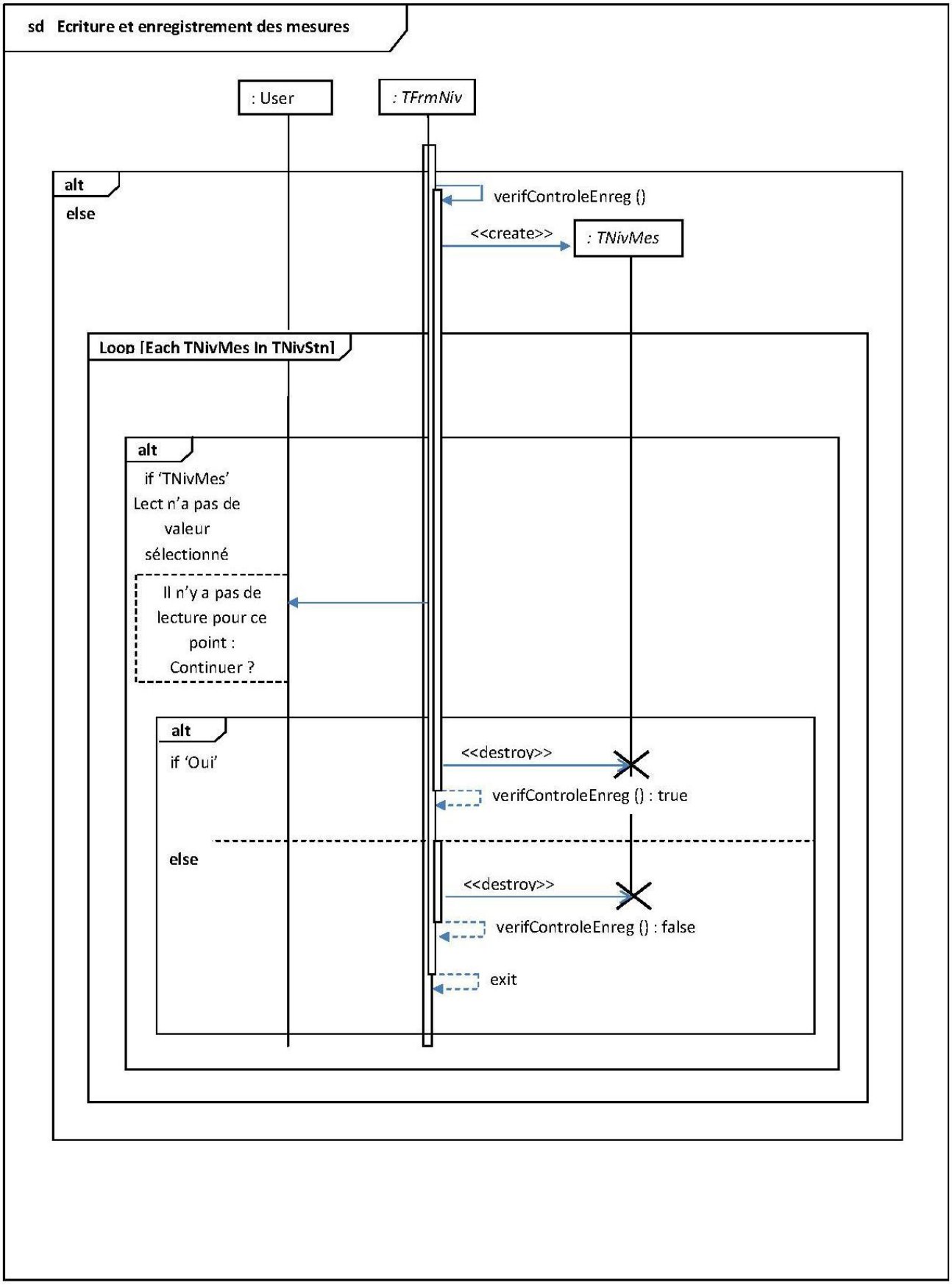


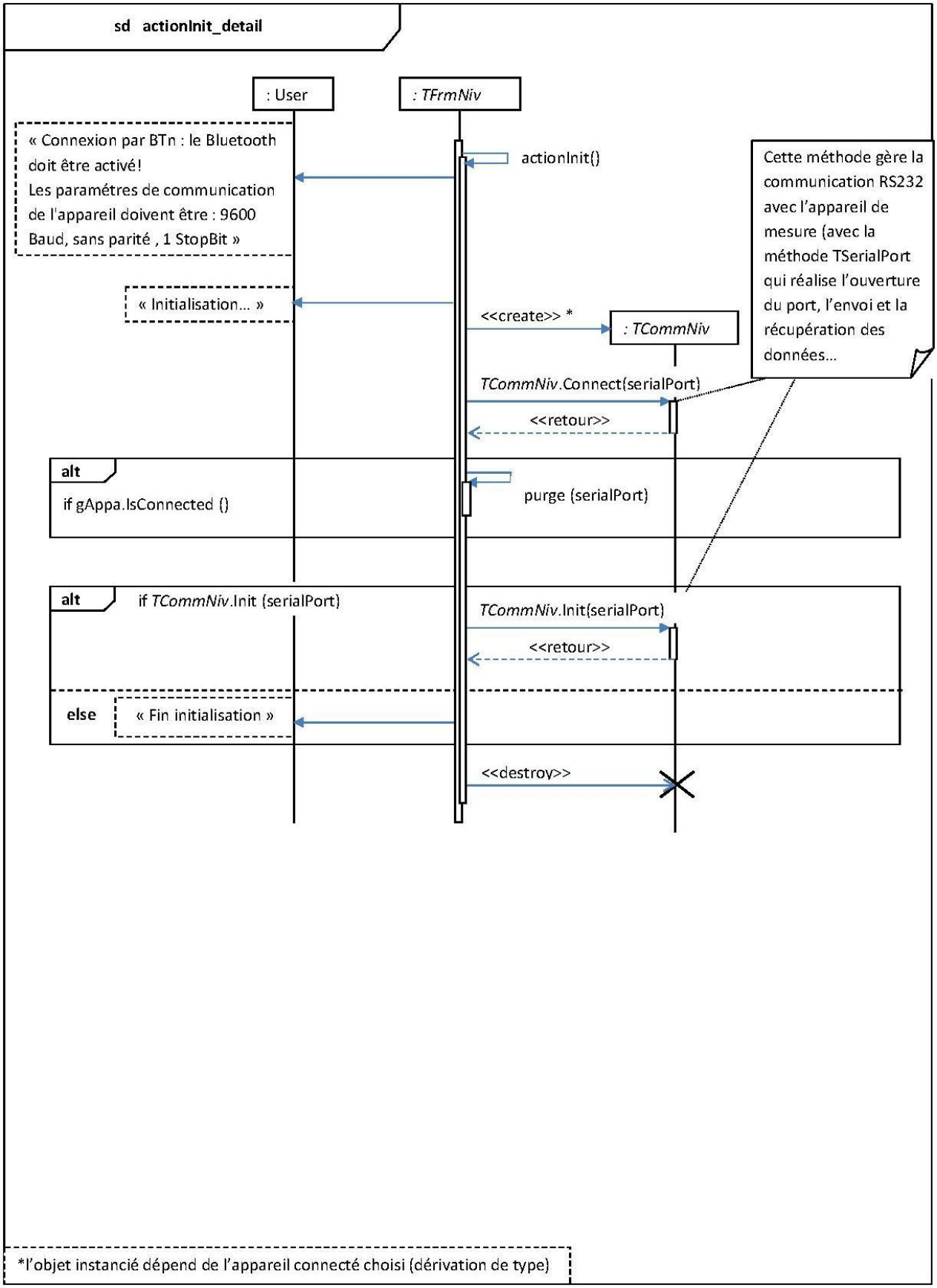
Annexe 16
Diagramme de séquence – Module de nivellement – Ecriture et enregistrement des mesures

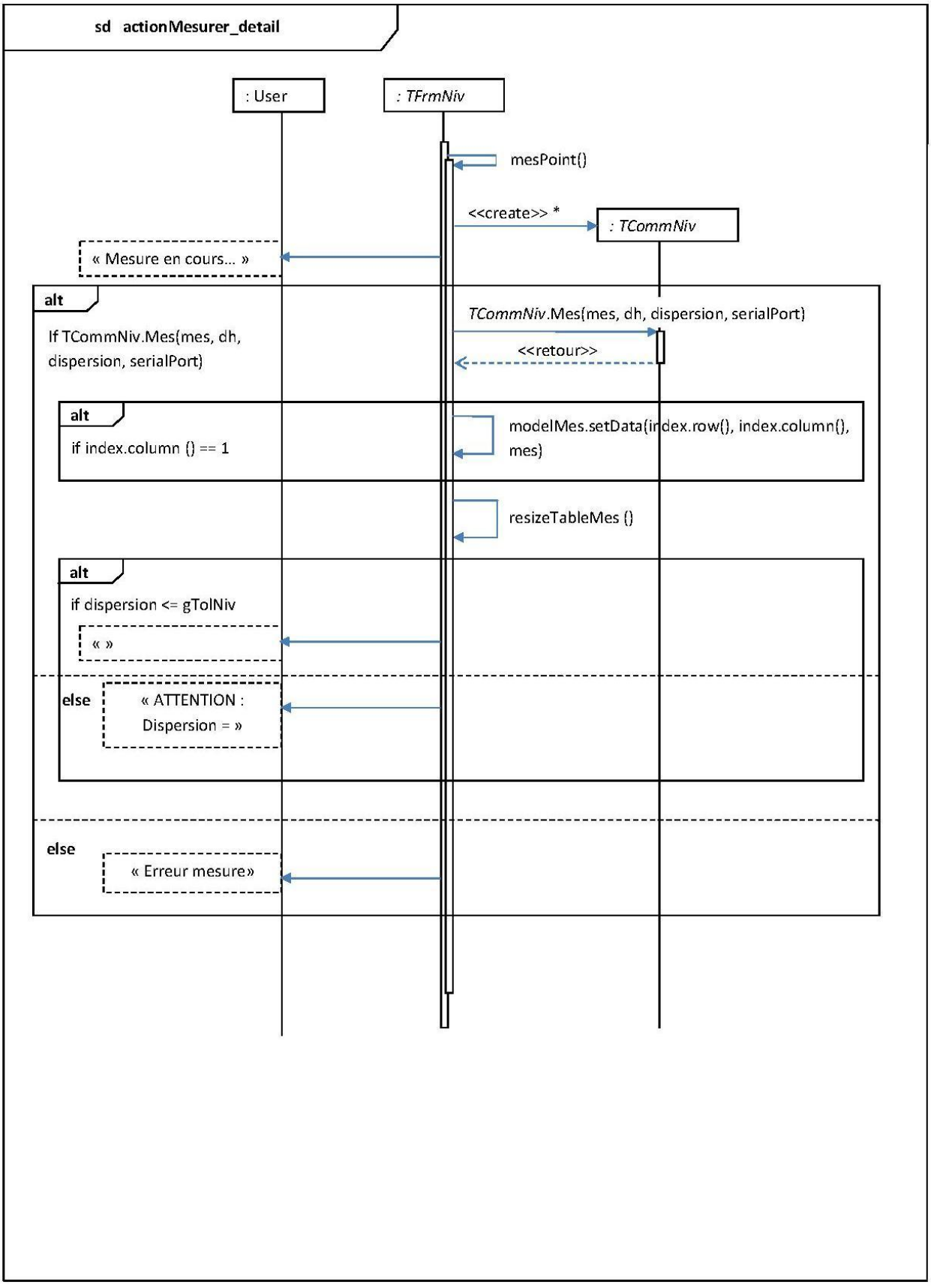



sd Ecriture et enregistrement des mesures
: User
: TFrmNiv
: TNivStations
Cette méthode charge la liste des points retour dans la variable globale gSequRetour
alt
if 'Retour' sélectionné
actionRetour ()
<<create>>
: TNivStation
gSequRetour .open (gRepTheo, "Niv_retour.txt")
Loop [Tous les TNivStation dans gSequRetour]
reverse ()
<<destroy>>
Cette méthode inverse chaque station de la variable gSequRetour
alt
if (appareil non connecté)
Sélection d'un point de mesure
startEditMes ()
Ecriture de la mesure
startEditMes ()
Choix de la mire si diff.
startEditMes ()
else
Sélection d'un point de mesure
startEditMes ()
Appui sur bouton 'Init'
actionInit ()
ref
actionInit_detail
Appui sur bouton 'Mesurer'
actionMesurer ()
ref
actionMesurer_detail
Choix de la mire si diff.
startEditMes ()

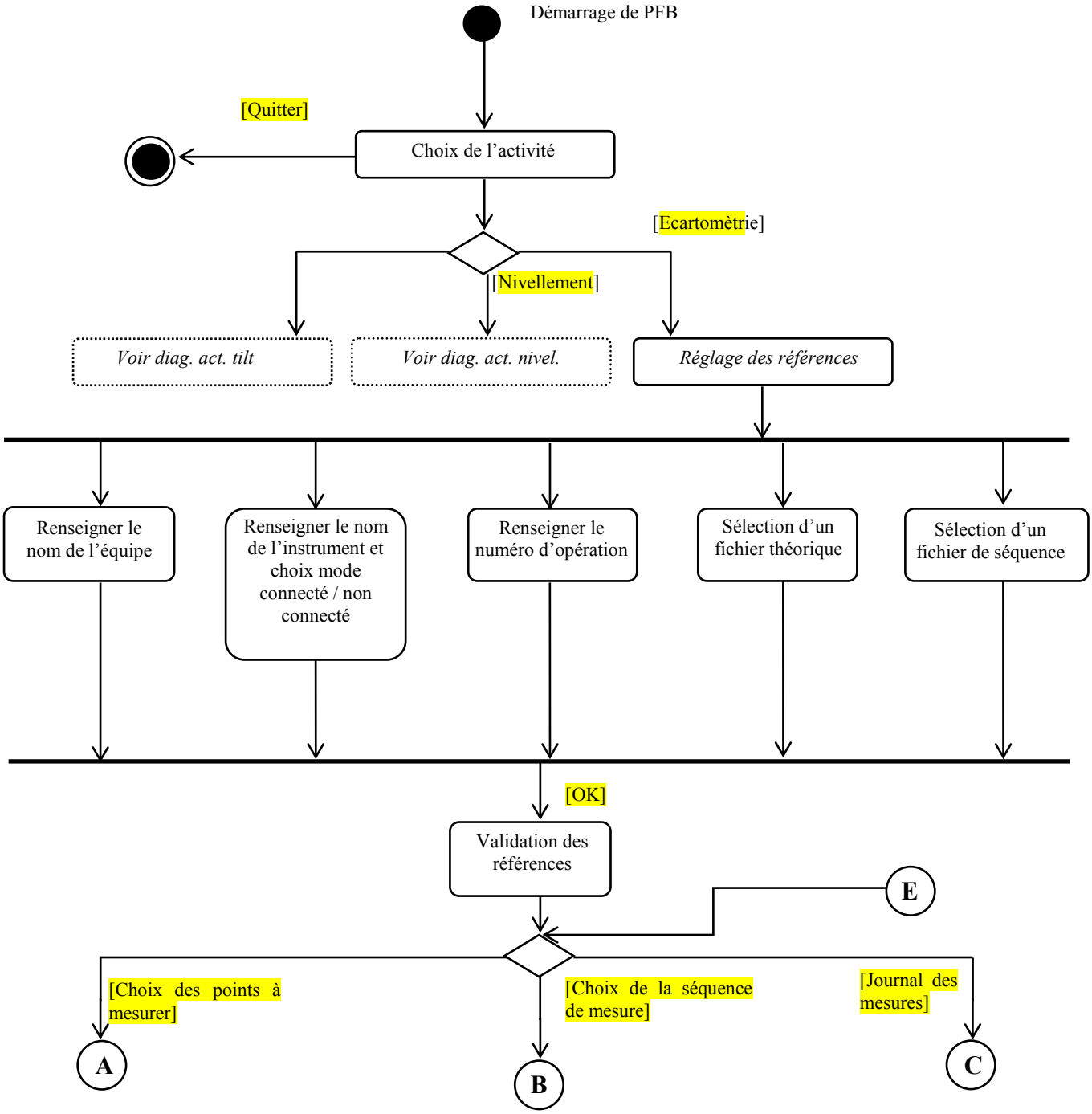

sd Ecriture et enregistrement des mesures
: User
: TFrmNiv
: TReference
: TNivStation*
Appui sur bouton 'Enregistrer'
butEnreg ()
actionEnregistrer ()
<<create>>
: TNivMes
Cette méthode vérifie s'il y a bien une lecture enregistré sur chaque point
alt
if verifControleEnreg ()
verifControleEnreg
Cette méthode calcul l'erreur moyenne quadratique sur les mesures réalisées
alt
if 'Retour' sélectionné
calculEMQ
Loop [Each TNivMes In TNivStn]
TNivMes.comment = ("emq"...)
ecrit geode REF(gRepMes)
Ces méthodes écrivent le fichier de sortie .dat
ecrit geode OBJ(gRepMes)
ecrit geode LE(gRepMes)
alt
if 'Retour' non sélectionné
Cette méthode écrit le fichier retour.txt
ecrit Retour(gRepTheo)
alt
if emq = horsTol
alt
if 'Oui'
Reprise de station Oui/Non
actionReprise ()
preparerStation (indSeq)
Cette méthode a pour but d'afficher la station suivante dans le modelChoix et modelMes à partir de la variable globale gSeqNivStation
*variable de classe

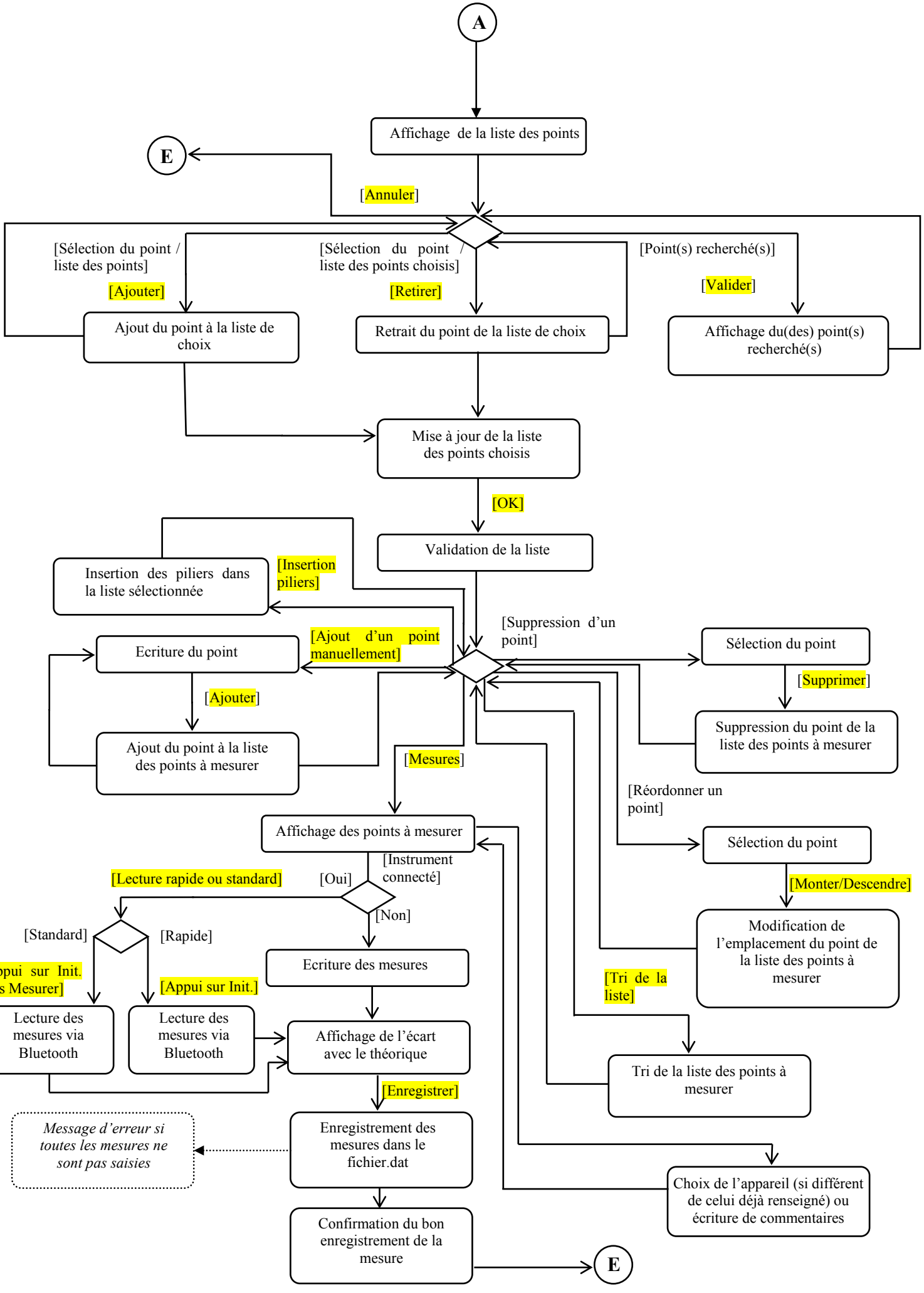

sd Ecriture et enregistrement des mesures
: User
: TFrmNiv
alt
else
verifControleEnreg ()
<<create>>
: TNivMes
Loop [Each TNivMes In TNivStn]
alt
if 'TNivMes' Lect n'a pas de valeur sélectionné
Il n'y a pas de lecture pour ce point : Continuer ?
alt
if 'Oui'
<<destroy>>
verifControleEnreg () : true
else
<<destroy>>
verifControleEnreg () : false
exit

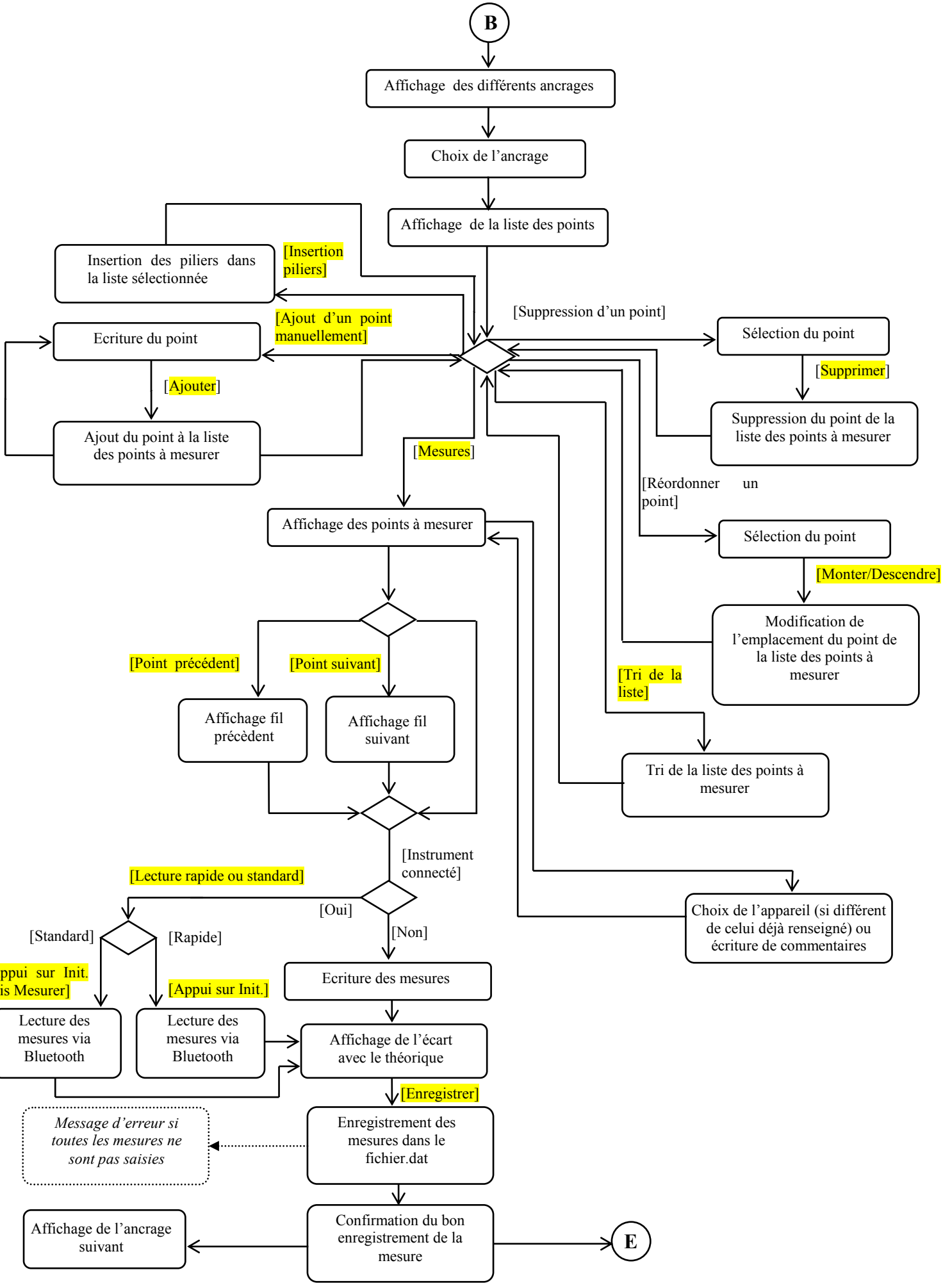

sd actionInit_detail
: User
: TFrmNiv
« Connexion par BTn : le Bluetooth doit être activé!
Les paramétres de communication de l'appareil doivent être : 9600 Baud, sans parité , 1 StopBit »
actionInit()
Cette méthode gère la communication RS232 avec l'appareil de mesure (avec la méthode TSerialPort qui réalise l'ouverture du port, l'envoi et la récupération des données...
« Initialisation... »
<<create>> *
: TCommNiv
TCommNiv.Connect(serialPort)
<<retour>>
alt
if gAppa.IsConnected ()
purge (serialPort)
alt
if TCommNiv.Init (serialPort)
TCommNiv.Init(serialPort)
<<retour>>
else
« Fin initialisation »
<<destroy>>
*l'objet instancié dépend de l'appareil connecté choisi (dérivation de type)

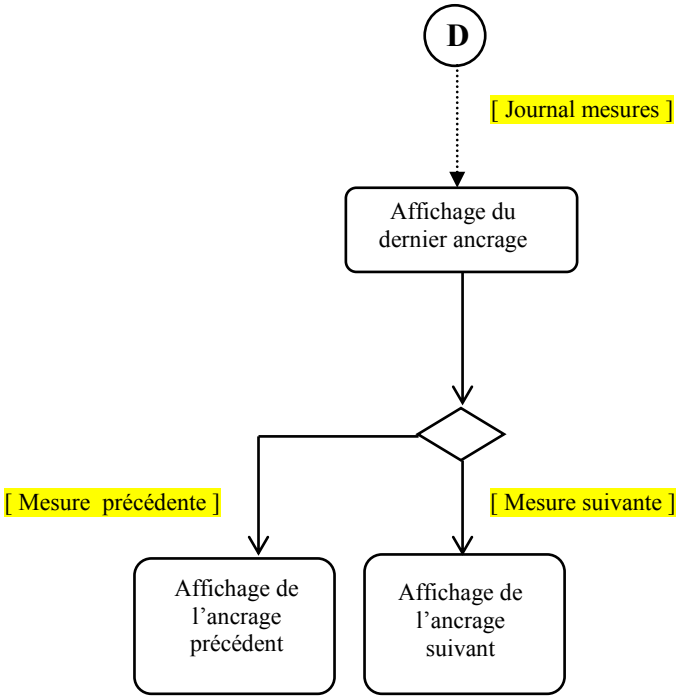

sd actionMesurer_detail
: User
: TFrmNiv
mesPoint()
<<create>> *
: TCommNiv
« Mesure en cours... »
alt
If TCommNiv.Mes(mes, dh, dispersion, serialPort)
TCommNiv.Mes(mes, dh, dispersion, serialPort)
<<retour>>
alt
if index.column () == 1
modelMes.setData(index.row(), index.column(), mes)
resizeTableMes ()
alt
if dispersion <= gTolNiv
« »
else
« ATTENTION : Dispersion = »
else
« Erreur mesure»

Annexe 17
Diagramme d'activité – Module d'écartométrie

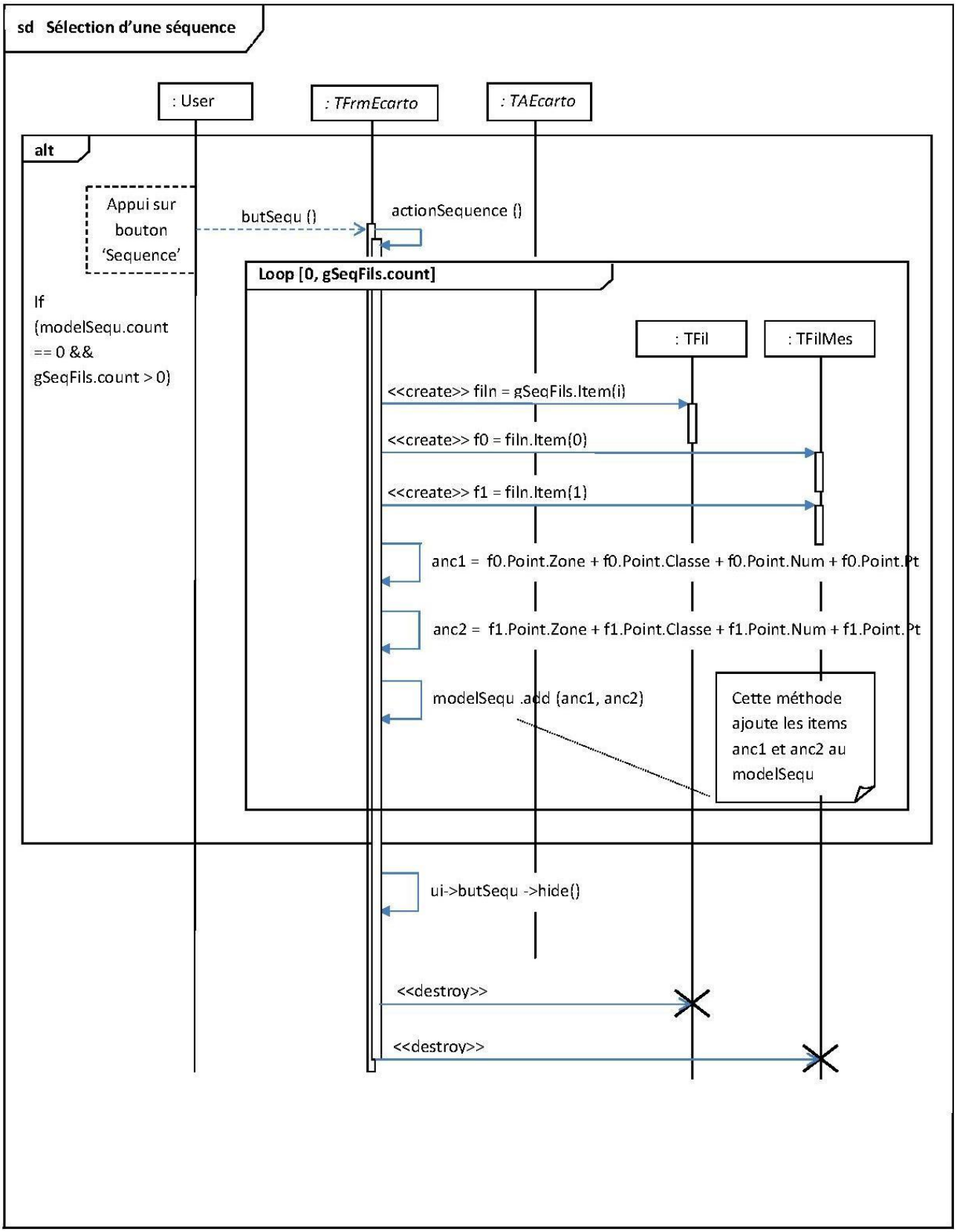


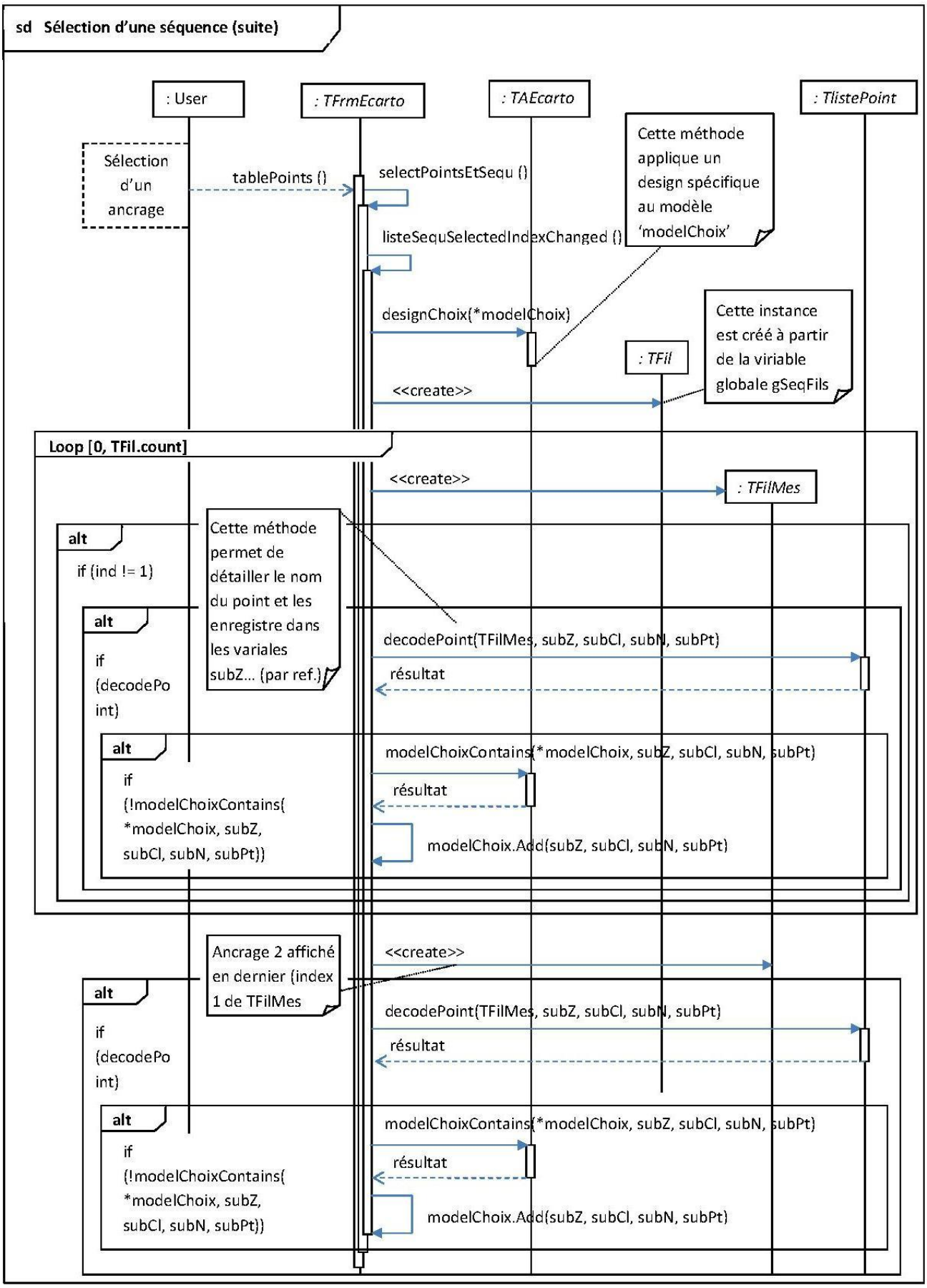

A
Affichage de la liste des points
E
[Annuler]
[Sélection du point / liste des points]
[Ajouter]
Ajout du point à la liste de choix
[Sélection du point / liste des points choisis]
[Retirer]
Retrait du point de la liste de choix
[Point(s) recherché(s)]
[Valider]
Affichage du(des) point(s) recherché(s)
Mise à jour de la liste des points choisis
[OK]
Validation de la liste
Insertion des piliers dans la liste sélectionnée
[Insertion piliers]
[Suppression d'un point]
Sélection du point
[Supprimer]
Suppression du point de la liste des points à mesurer
Ecriture du point
[Ajout d'un point manuellement]
[Ajouter]
Ajout du point à la liste des points à mesurer
[Mesures]
[Réordonner un point]
Affichage des points à mesurer
Sélection du point
[Monter/Descendre]
Modification de l'emplacement du point de la liste des points à mesurer
[Instrument connecté]
[Oui]
[Non]
[Lecture rapide ou standard]
[Standard]
[Rapide]
[Appui sur Init. puis Mesurer]
[Appui sur Init.]
Ecriture des mesures
[Tri de la liste]
Tri de la liste des points à mesurer
Lecture des mesures via Bluetooth
Lecture des mesures via Bluetooth
Affichage de l'écart avec le théorique
[Enregistrer]
Message d'erreur si toutes les mesures ne sont pas saisies
Enregistrement des mesures dans le fichier.dat
Choix de l'appareil (si différent de celui déjà renseigné) ou écriture de commentaires
Confirmation du bon enregistrement de la mesure
E


B
Affichage des différents ancrages
Choix de l'ancrage
Affichage de la liste des points
Insertion des piliers dans la liste sélectionnée
[Insertion piliers]
[Ajout d'un point manuellement]
Ecriture du point
[Ajouter]
Ajout du point à la liste des points à mesurer
[Suppression d'un point]
Sélection du point
[Supprimer]
Suppression du point de la liste des points à mesurer
[Mesures]
[Réordonner un point]
Affichage des points à mesurer
Sélection du point
[Monter/Descendre]
Modification de l'emplacement du point de la liste des points à mesurer
[Point précédent]
[Point suivant]
[Tri de la liste]
Affichage fil précèdent
Affichage fil suivant
Tri de la liste des points à mesurer
[Instrument connecté]
[Lecture rapide ou standard]
[Oui]
[Non]
Choix de l'appareil (si différent de celui déjà renseigné) ou écriture de commentaires
[Standard]
[Rapide]
[Appui sur Init. puis Mesurer]
[Appui sur Init.]
Ecriture des mesures
Lecture des mesures via Bluetooth
Lecture des mesures via Bluetooth
Affichage de l'écart avec le théorique
[Enregistrer]
Message d'erreur si toutes les mesures ne sont pas saisies
Enregistrement des mesures dans le fichier.dat
Affichage de l'ancrage suivant
Confirmation du bon enregistrement de la mesure
E

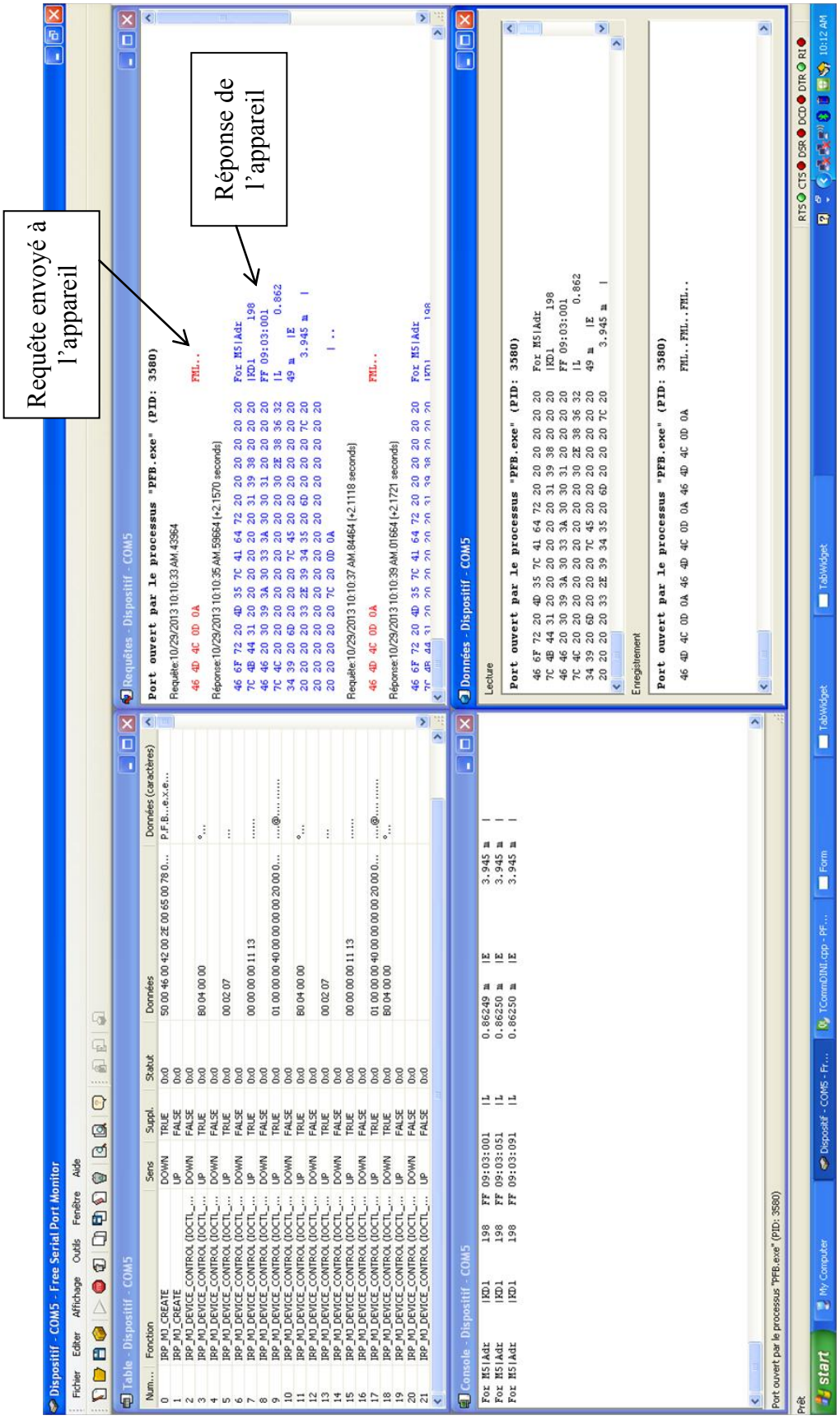

D
[Journal mesures]
Affichage du dernier ancrage
[Mesure précédente]
[Mesure suivante]
Affichage de l'ancrage précédent
Affichage de l'ancrage suivant

Annexe 18
Diagramme de séquence – Module d'écartométrie – Sélection d'une séquence

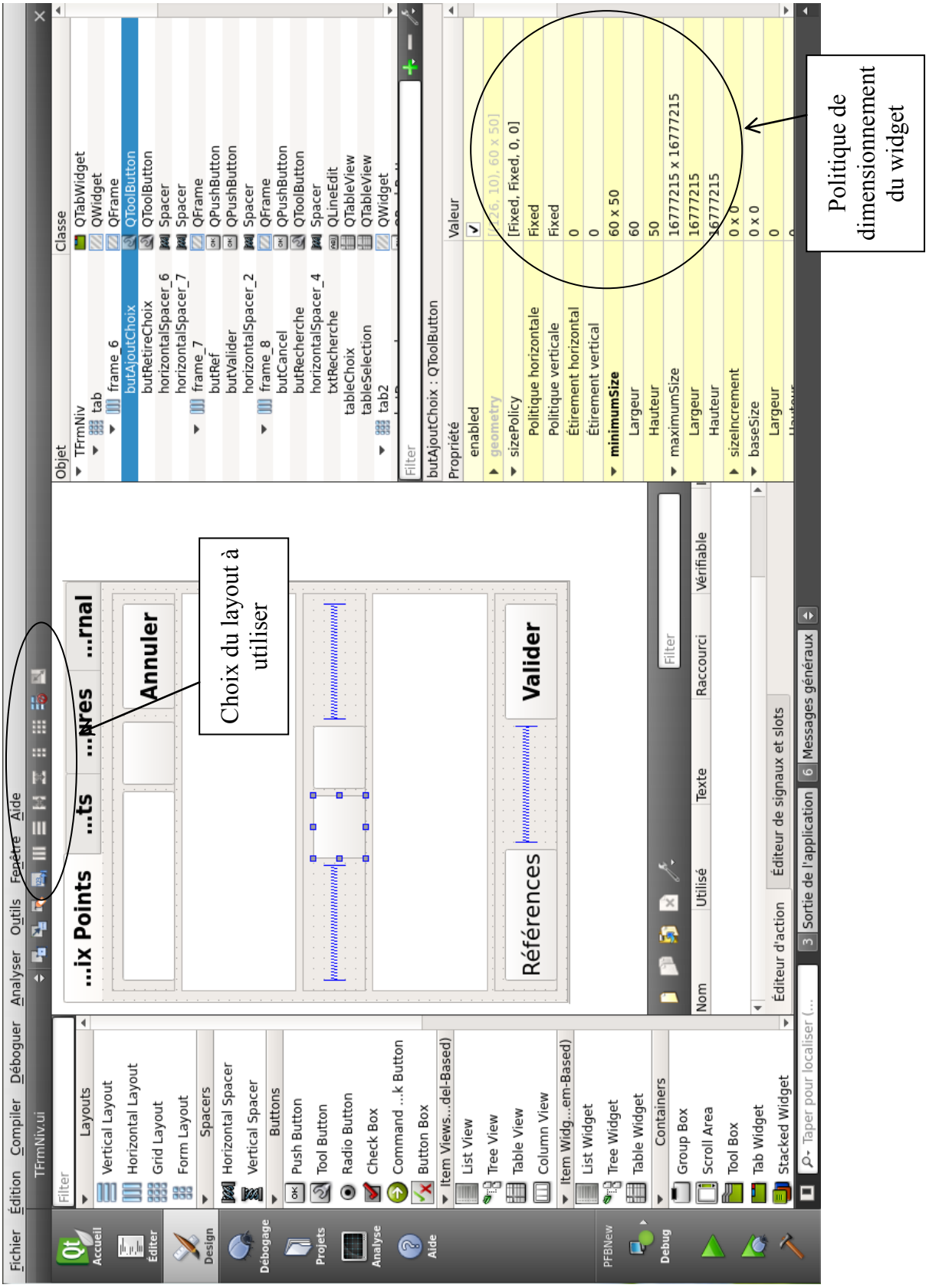



sd Sélection d'une séquence (suite)
: User
: TFrmEcarto
: TAEcarto
: TlistePoint
Sélection d'un ancrage
tablePoints ()
selectPointsEtSequ ()
Cette méthode applique un design spécifique au modèle 'modelChoix'
listeSequSelectedIndexChanged ()
designChoix(*modelChoix)
: TFil
Cette instance est créé à partir de la viriable globale gSeqFils
<<create>>
Loop [0, TFil.count]
<<create>>
: TFilMes
alt
if (ind != 1)
Cette méthode permet de détailler le nom du point et les enregistre dans les variales subZ... (par ref.)
alt
if (decodePoint)
decodePoint(TFilMes, subZ, subCl, subN, subPt)
résultat
alt
if (!modelChoixContains(*modelChoix, subZ, subCl, subN, subPt))
modelChoixContains(*modelChoix, subZ, subCl, subN, subPt)
résultat
modelChoix.Add(subZ, subCl, subN, subPt)
Ancrage 2 affiché en dernier (index 1 de TFilMes
<<create>>
alt
if (decodePoint)
decodePoint(TFilMes, subZ, subCl, subN, subPt)
résultat
alt
if (!modelChoixContains(*modelChoix, subZ, subCl, subN, subPt))
modelChoixContains(*modelChoix, subZ, subCl, subN, subPt)
résultat
modelChoix.Add(subZ, subCl, subN, subPt)

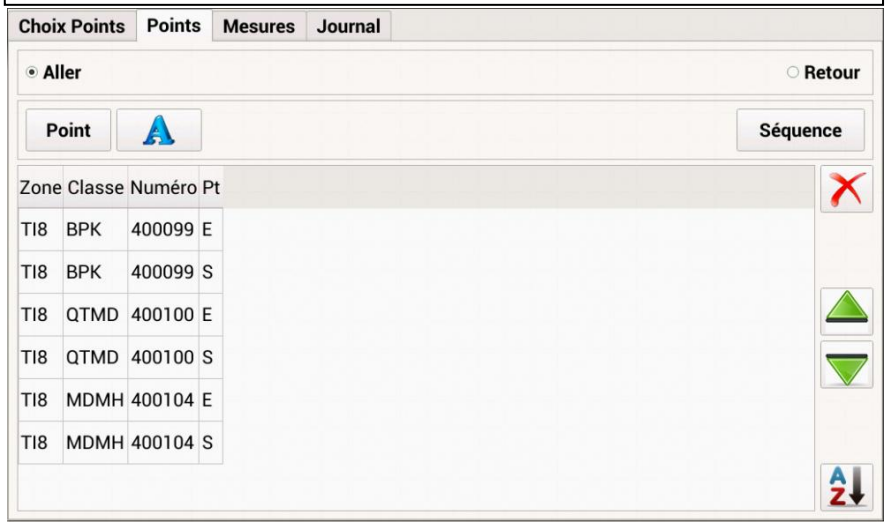
Annexe 19
Free Serial Port Monitor



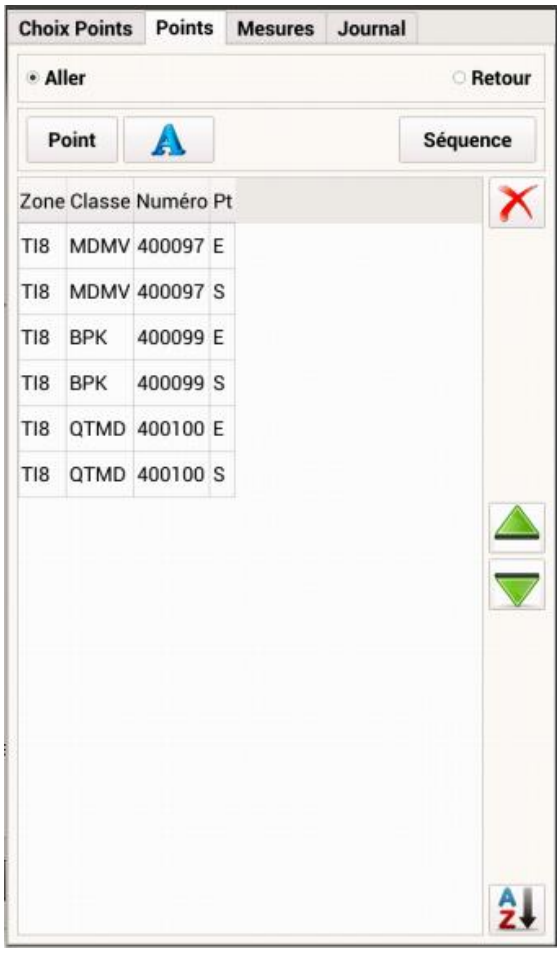
Annexe 20
Utilisation des layouts sous Qt Designer



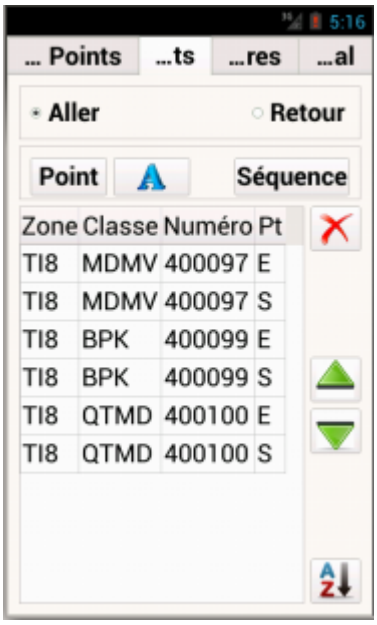
Affichage en mode paysage sur une tablette 7 pouces – Définition 600 x 1024



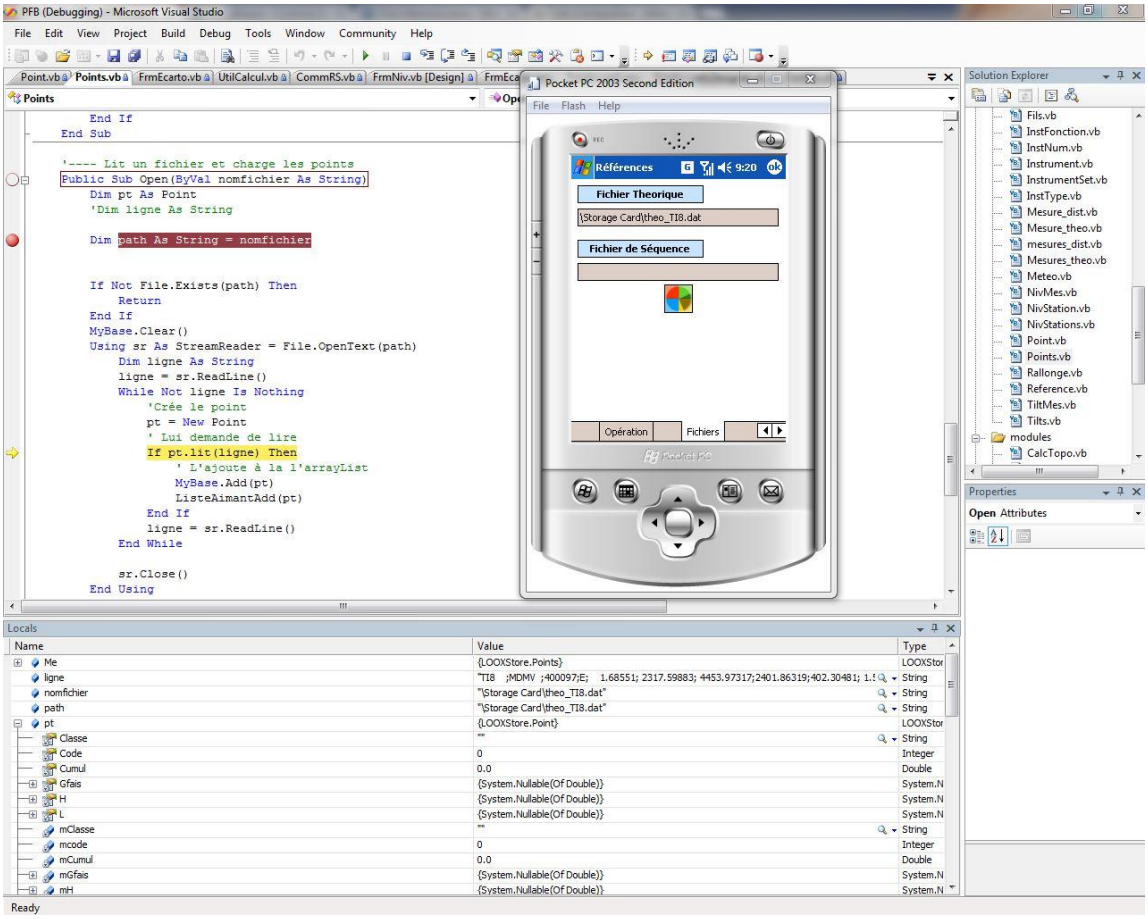
Affichage en mode portrait sur une tablette 7 pouces – Définition 1024 x 600

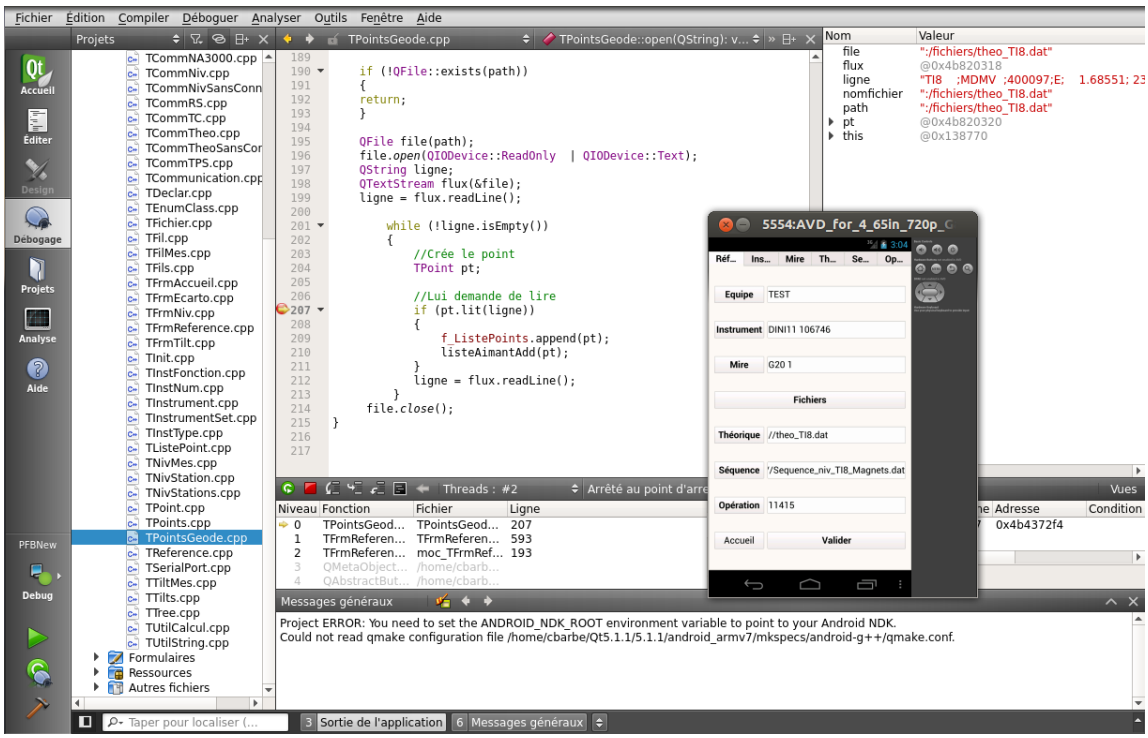


Affichage en mode portait sur un Smartphone de 3,7 pouces – Définition 800 x 480



Annexe 21
Tests d'intégration continus : comparaison des deux versions de PFB





Annexe 22
Correction du problème de multiples enregistrements

```
bool TFrmNiv::verifControleEnreg()
{
    TNivMes *nivi = new TNivMes();
    int ret = 1024;

    for (int i = 0; i < nivStn->listeNivMes().count(); i++)
    {
        *nivi = nivStn->listeNivMes().value(i);
        if (nivi->lect() == -99999999)
        {
            QDesktopWidget *desktop = new QDesktopWidget;
            int largeur = desktop->screenGeometry().width();
            int hauteur = desktop->screenGeometry().height();

            QMessageBox msgBox;
            QFont font("Tahoma", 16, QFont::Bold);
            msgBox.setText("ATTENTION");
            msgBox.setInformativeText("Il n'y a pas de lecture pour ce point : "
+ nivi->point().nom());
            msgBox.setBaseSize(250, 150);
            msgBox.setMaximumSize(largeur, hauteur);
            msgBox.setFont(font);
            msgBox.setStandardButtons(QMessageBox::Ok | QMessageBox::Cancel);
            ret = msgBox.exec();

        }

        if (ret == 1024 && nivi->lect() == -99999999)
        {
            nivStn->removeAt(i);
            modelMes->removeRow(i + 1);
        }
    }

    if(ret != 1024)
    {
        return false;
    }

    return true;

}
```

Annexe 23
Nivellement - Affichage du journal

```
void TFrmNiv::afficheJournal()
{

    if (indSeqJournAffich >= 0 && indSeqJournAffich <= indSeqJourn)
    {
    int indexAffiche = indSeqJournAffich * 5;

    QVector<QString> nomPoints = f_ListeNivRecord->value(indexAffiche);
    QVector<QString> lectures = f_ListeNivRecord->value(indexAffiche + 1);
    QVector<QString> rallonges = f_ListeNivRecord->value(indexAffiche + 2);
    QVector<QString> ecart = f_ListeNivRecord->value(indexAffiche + 3);
    QVector<QString> mire = f_ListeNivRecord->value(indexAffiche + 4);

    int nb = nomPoints.count();

    modelJour->clear();

    for (int i = 0;  i < nb; i++)
    {
        QStandardItem *item = new QStandardItem(nomPoints.value(i));
        item->setFlags(item->flags() & ~Qt::ItemIsEditable);
        modelJour->setItem(i+1, 0, item);
        QStandardItem *item2 = new QStandardItem(lectures.value(i));
        item2->setFlags(item2->flags() & ~Qt::ItemIsEditable);
        modelJour->setItem(i+1, 1, item2);
        QStandardItem *item3 = new QStandardItem(rallonges.value(i));
        item3->setFlags(item3->flags() & ~Qt::ItemIsEditable);
        modelJour->setItem(i+1, 2, item3);
        QStandardItem *item4 = new QStandardItem(ecart.value(i));
        item4->setFlags(item4->flags() & ~Qt::ItemIsEditable);
        modelJour->setItem(i+1, 3, item4);
        QStandardItem *item5 = new QStandardItem(mire.value(i));
        item5->setFlags(item5->flags() & ~Qt::ItemIsEditable);
        modelJour->setItem(i+1, 4, item5);
    }

    ui->txtEnregJour->setText(f_ListeNivRecordInfo->value(indSeqJournAffich))

    }

   designMes(*modelJour);
   resizeTableJour();
}
```

Annexe 24
Documentation générée par Doxygen

```cpp
#ifndef TAECARTO_H
#define TAECARTO_H

#include <QStandardItemModel>
#include <QString>

//! Classe TAEcarto.
/*!
Cette classe est abstraite, elle est hérité par la classe TFrmEcarto.
Le but de cette classe est de gérer les données manipulées dans la classe TFrmEcarto et de réaliser les designs des tableaux.
*/

class TAEcarto

{

public:
    TAEcarto();
    ~TAEcarto();
    /*!
     * Cette méthode ajoute les points du fichier théo au modèle qui est affiché dans le tableau tableChoix
     */
    void fillList(QStandardItemModel &model);
    /*!
     * Cette méthode est utilisé par la fonction recherche de l'onglet 'Choix Points'
     * L'argument 'nom' est la valeur recherchée par l'utilisateur et le modèle passé en argument 'model'
     * est celui qui va être modifié pour afficher le résultat de la recherche
     */
    void searchPt(QString const& nom, QStandardItemModel &model);
    /*!
     * Cette méthode vérifie l'existance d'un point donné par l'argument 'nom' dans le modèle passé en argument 'model'
     */
    bool searchItem(QString const& nom, QStandardItemModel &model);
    /*!
     * Cette méthode retourne le nombre de ligne présent dans le modèle passé en argument 'model'
     */
    int nbRow(QStandardItemModel &model);
    /*!
     * Cette méthode retourne l'index de l'argument 'nom' de la variable globale gPointsGeode
     */
    int indexPt(QString const& nom);
    /*!
     * Cette méthode copie les données présentes dans le modèle passé en argument 'model' dans le modèle passé en argument 'model
     */
```

Detailed Description

Classe **TAEcarto**.

Cette classe est abstraite, elle est hérité par la classe **TFrmEcarto**. Le but de cette classe est de gérer les données manipulées dans la classe **TFrmEcarto** et de réaliser les designs des tableaux.

Member Function Documentation

void TAEcarto::copyModelSelect (QStandardItemModel & model,
QStandardItemModel & model2
)

Cette méthode copie les données présentes dans le modèle passé en argument 'model' dans le modèle passé en argument 'model2'

void TAEcarto::designChoice (QStandardItemModel & model)

Cette méthode applique un design spécifique au modèle passé en argument 'model' Ce modèle est utilisé pour gérer l'affichage des données dans le tableau figurant dans l'onglet 'Points'

void TAEcarto::designMes (QStandardItemModel & model)

Cette méthode applique un design spécifique au modèle passé en argument 'model' Ce modèle est utilisé pour gérer l'affichage des données dans le tableau figurant dans l'onglet 'Mesures'

void TAEcarto::designSequ (QStandardItemModel & model)

Cette méthode applique un design spécifique au modèle passé en argument 'model' Ce modèle est utilisé pour gérer l'affichage des données dans le tableau figurant dans l'onglet 'Points' pour l'affichage des séquences

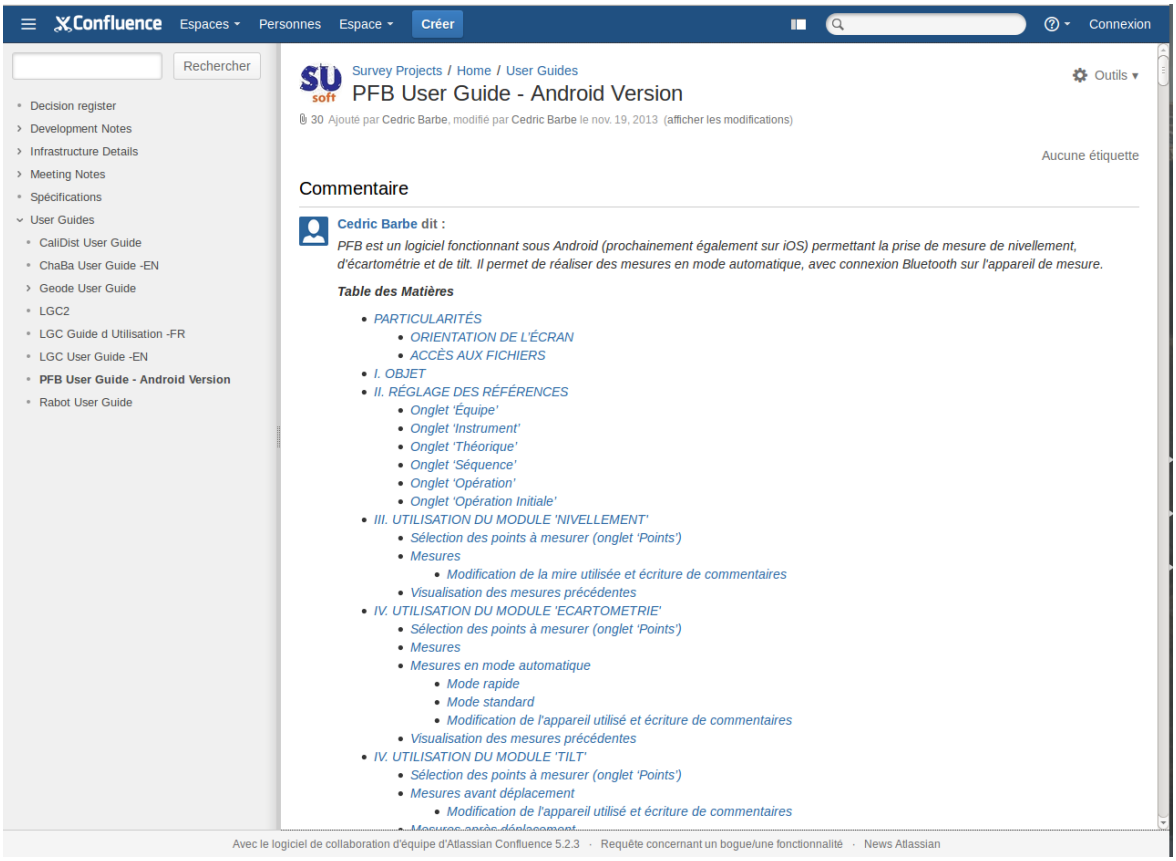
void TAEcarto::fillList (QStandardItemModel & model)

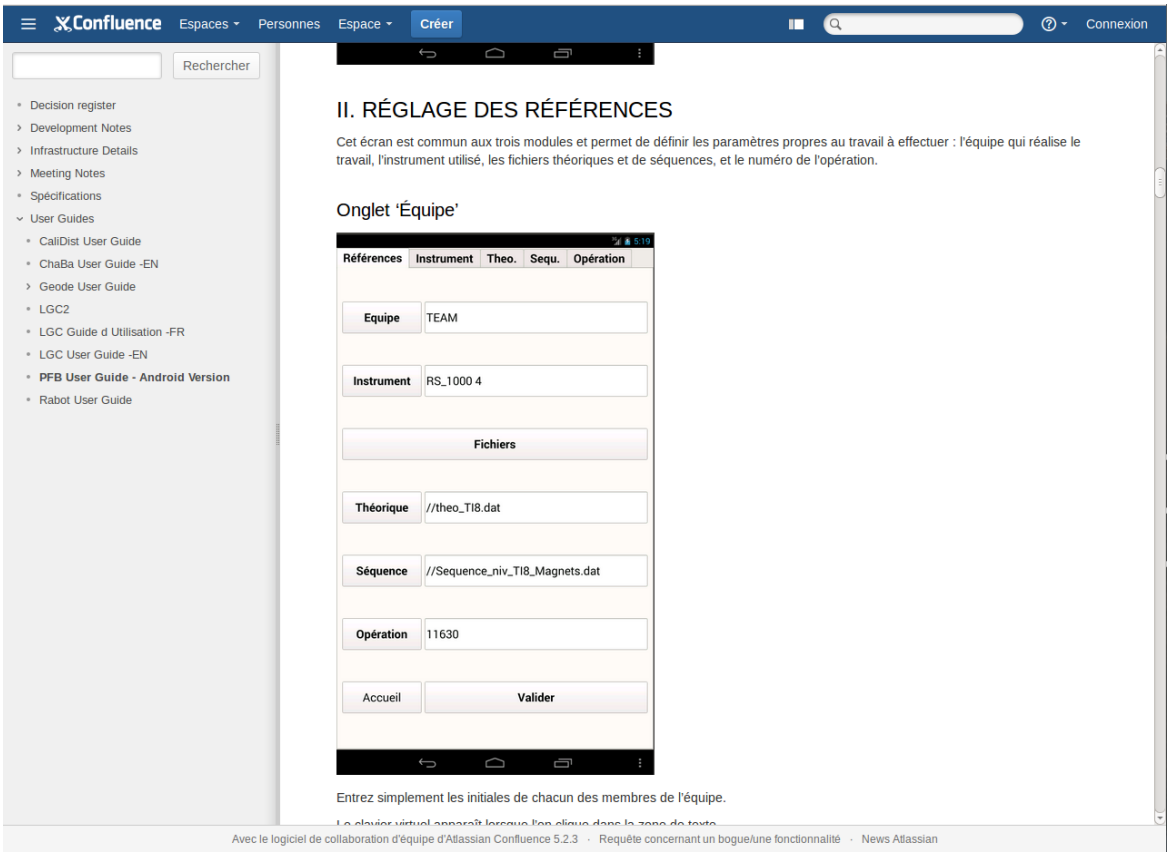
Cette méthode ajoute les points du fichier théo au modèle qui est affiché dans le tableau tableChoix

void TAEcarto::fillMes (QString const & nom,
QStandardItemModel & model
)

Cette méthode returne le nombre de ligne présent dans le modèle passé en argument 'model'

Annexe 25
Documentation utilisateur sur Confluence





Liste des figures

Figure 1 : Le Grand collisionneur de hadrons 24
Figure 2 : schéma représentant les différents accélérateurs fonctionnant au CERN 25
Figure 3 : Le détecteur CMS 26
Figure 4 : Le système de coordonnées du CERN et de l'orbite PS 30
Figure 5 : Mesure de nivellement sur un dipôle du LHC 33
Figure 6 : Stations de mesures 34
Figure 7 : Fils de mesures 34
Figure 8 : Menu principal de PFB 39
Figure 9 : Réglage des références - Module de nivellement, d'écartométrie et tilt 40
Figure 10 : Choix de l'opération 41
Figure 11 : Choix de l'instrument 41
Figure 12 : Choix de la mire 41
Figure 13 : Module de nivellement - Sélection des points 42
Figure 14 : Choix des points 42
Figure 15 : Choix d'une séquence (station) 42
Figure 16 : Ajout manuel d'un point 42
Figure 17 : Mesures de tilt 43
Figure 18 : Mesures de nivellement 43
Figure 19 : Mesures d'écartométrie 43
Figure 20 : Prise de mesures - Data Workflow 45
Figure 21 : Calcul du facteur d'ajustement 58
Figure 22 : Calcul de la charge de travail 58
Figure 23 : Extrait du diagramme d'activité - Module de tilt 61
Figure 24 : Extrait du diagramme de séquences 62
Figure 25 : Extrait du diagramme d'activité - Module de nivellement 63
Figure 26 : Extrait du diagramme de séquences 66
Figure 27 : Extrait du diagramme d'activité - Module d'écartométrie 67
Figure 28 : Extrait du diagramme de séquences 68
Figure 29 : Implémentation QDirModel 69
Figure 31 : Connexion signal/slot pour adaptation tactile 70
Figure 30 : Ecran de sélection d'un fichier théorique 70
Figure 32 : Slot pour adaptation tactile 71
Figure 33 : Adaptateur Bluetooth/RS232 71
Figure 34 : Méthode 'SendMes' de la classe TCommDINI 71
Figure 35 : Portion de code - Calcul de l'emq - VB.Net 74
Figure 36 : Portion de code - Calcul de l'emq - C++ 74
Figure 37 : Nivellement - Saisie des mesures 75
Figure 38 : Nivellement - Copie des mesures au journal 75

Liste des tableaux

Tableau 1 : Données figurant dans un fichier théorique 40
Tableau 2 : Données figurant dans un fichier séquence de nivellement 40
Tableau 3 : Données figurant dans un fichier séquence d'écartométrie 41
Tableau 4 : Module de nivellement - Exemple de fichier de sortie .dat 44
Tableau 5 : Module de nivellement - Exemple de fichier 'Retour.txt' 44
Tableau 6 : Module d'écartométrie - Exemple de fichier de sortie .dat 45
Tableau 7 : Module de tilt - Exemple de fichier de sortie .dat 45

Évolution des logiciels permettant de gérer l'alignement des électroaimants supraconducteurs du LHC

Mémoire d'Ingénieur C.N.A.M., Lyon 2014

RESUME

Après l'étude de la problématique qui était principalement lié au choix de la solution multiplateforme, la réécriture du logiciel destiné à la prise de mesure a été réalisé, afin de garantir son succès, selon les fondamentaux du génie logiciel. Il en a découlé un planning de développement, la création des documents liés à la conception générale et détaillée.

Grâce au logiciel existant, les tests ont été facilités et cela a garantie le bon fonctionnement du nouveau logiciel. La maintenance de celui-ci est maintenant facilité grâce à la documentation et Qt rempli les deux objectifs, le logiciel est multiplateforme et les données sont séparées de la vue.

Mots clés : Qt, multiplateforme, Android, génie logiciel.

SUMMARY

After studying the issue, which was mainly due to the choice of multiplatform solution, rewriting the software for measurements has been made to ensure its success, according to the fundamentals of software engineering. This has resulted in a development planning, the creation of documents related to general and detailed design.

With the existing software, the tests have been facilitated and guarantee the proper functioning of the new software. Maintenance it is now easier with the documentations and Qt completed two objectives, the new software is multiplatform and data are separated from the view.

Key words : Qt, multiplatform, Android, software engineering.