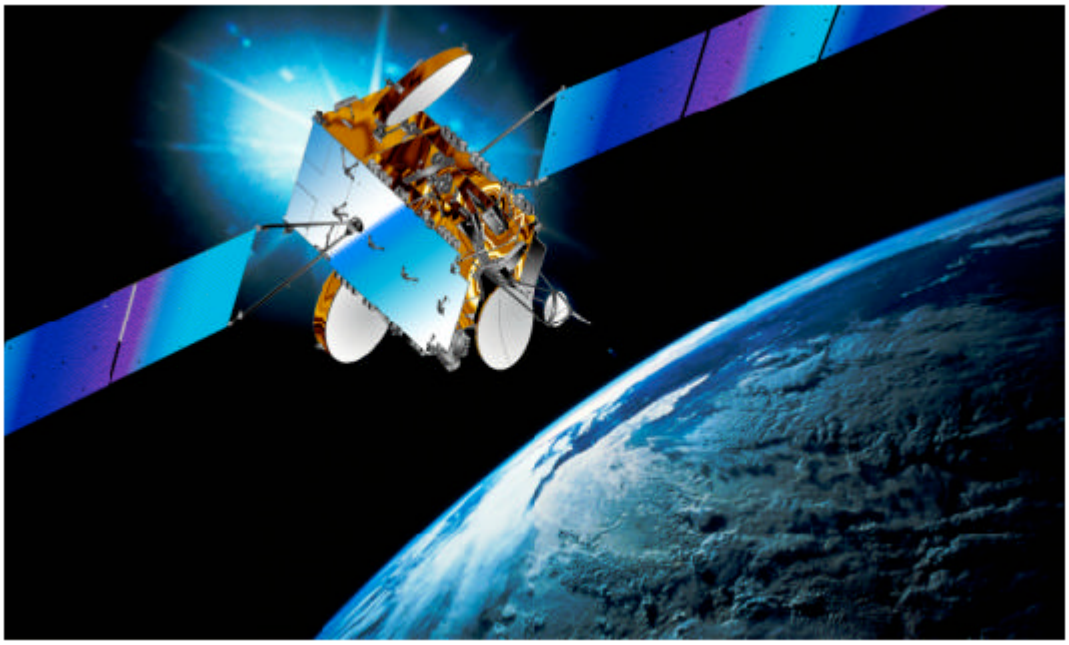
Jérôme BRAURE

Rapport de Stage

Participation à un développement logiciel



04.04.2005 – 23.09.2005

© Alcatel Alenia Space 2005

© Alcatel Alenia Space 2005

© Alcatel Alenia Space 2005

1 Introduction

Ce document constitue le rapport de stage effectué dans le cadre du « Mastère spécialisé en Techniques Aéronautiques et Spatiales » option « Astronautique : satellites et systèmes » dispensé à l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (ENSAE / SUPAERO), en partenariat avec l'Ecole Nationale Supérieure d'Ingénieurs de Constructions Aéronautiques (ENSICA) de Toulouse.

Le stage s'est déroulé sur le site d'Alcatel Space à Cannes du 4 avril au 23 septembre 2005, au sein du département Ligne de Produits Electroniques et Logiciels. L'objet du stage est une participation au développement du logiciel de vol (en particulier les tests unitaires) de KoreaSat V, un satellite du type SpaceBus 4000.

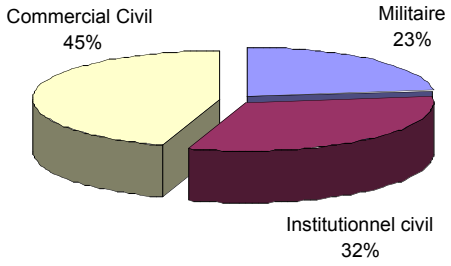
La structure du rapport est la suivante :

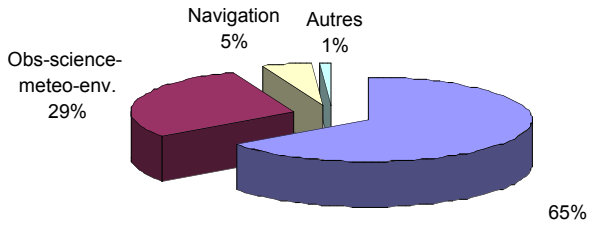
La section 2 présente la société Alcatel Space, ses activité et ses produits puis la section 3 présente le type de satellite Spacebus 4000, dont le développement du logiciel de vol constitue le sujet du stage. La section 4 donne un aperçu du département d'accueil du stage. Le processus de développement est ensuite présenté dans la section 5. La section 6 explique ce qui a été réalisé durant le stage et la section 7 la conclusion.

© Alcatel Alenia Space 2005

2 Alcatel Space

2.1 Activités





Alcatel Space est l'un des leaders mondiaux dans le domaine spatial (top 3) et le premier fournisseur mondial de satellites de télécommunications. Avec un savoir-faire dans les applications civiles et militaires, Alcatel Space développe des solutions de bout-en-bout pour les télécommunications, la navigation, la gestion des risques et de l'environnement, l'observation radar et optique, la météorologie et les sciences. Premier fournisseur spatial du Ministère français de la Défense, la société est également leader européen des satellites Dans les domaines des télécommunications, de la navigation, l'observation, la météorologie, et les sciences grâce à une gamme de :

- *Plates- formes satellites : Spacebus, Proteus, Météosat*
- *Charges utiles & instruments*
- *Equipements : micro-ondes, hyperfréquence, électronique, optique, radar, mécanismes,*
- *Structures, contrôle thermique.*
- *Segment sol : centres de contrôle, centres de mission, stations sol.*
- *Solutions réseaux : opérateurs fixes et mobiles, entreprises, institutions.*

2.2 Implantations et activités par site

En décembre 2004, Alcatel Space comprenait 5 540 collaborateurs. Le siège social d'Alcatel Space est à Paris et compte 5 établissements : Cannes, Kourou, Colombes, Toulouse et Valence. Les sites d'Alcatel Space ont des activités complémentaires notamment dans la conception et la fabrication de satellites.

***Colombes** : maîtrise d'œuvre de systèmes, de programmes militaires, de segments sol, des stations terriennes et des stations de connexion, des systèmes de télécommunications mais aussi des réseaux qui y sont associés, l'étude et le développement des produits et des systèmes de segments sol.*

***Cannes** : maîtrise d'œuvre de systèmes spatiaux complets. L'ingénierie des grands programmes satellites est effectuée par les équipes de la Direction des télécommunications spatiales, la réalisation et l'intégration de plateformes et d'instruments optiques et scientifiques, le traitement des images, la maîtrise d'œuvre de segments de contrôle satellites et des équipements pour satellite et segment sol.*

© Alcatel Alenia Space 2005

***Valence** : études, développement et productions de prototypes, de petites et moyennes séries pour les équipements électroniques embarqués ; conception, développement et production de circuits hybrides et MCM (Multi Chip Module) assurant des fonctions analogiques, numériques, de puissance et mixtes.*

***Toulouse** : Les charges utiles ou les équipements des satellites sont conçus et fabriqués sur le site : la maîtrise d'œuvre et l'intégration de systèmes de télécommunications de bout en bout, la maîtrise d'œuvre des systèmes sol de télécommunications et de navigation ainsi que celle des produits pour les segments sol, la réalisation et l'intégration de charges utiles et d'instruments radar hyperfréquences.*

***Kourou** : exploitation et maintenance des moyens sol pour la base spatiale (télécommunications notamment), support technique pour la mise à poste et le maintien en orbite des satellites.*

Alcatel Space compte également 4 filiales en Europe qui sont spécialisées dans des équipements ou produits complémentaires de ceux d'Alcatel Space pour les équipements bord. L'Industrial Unit Sol possède également deux filiales industrielles :

***Alcatel Bell Space**, à Hoboken, près d'Anvers : conçoit et fabrique notamment, des équipements de test électriques, des systèmes de communications, des "gateways et modems" pour le contrôle d'accès, mais aussi des systèmes de contrôle et des stations terrestres pour l'acquisition de données des satellites d'observation. La société comprend environ 75 personnes.*

***Alcatel Space Operations**, en Allemagne : Les 80 personnes sont en contrat chez les clients, notamment l'ESOC (site allemand de l'ESA qui contrôle les satellites). Elles effectuent notamment : les opérations et la mise à poste de satellites l'exploitation de stations sol Alcatel Space Netherlands réalise aux Pays-Bas des tests de matériels et d'activités d'assistance technique en télécommunications sur le site hollandais de l'ESA.*

Les deux filiales suivantes font partie de l'Industrial Unit Bord :

- ***Alcatel ETCA** : 440 personnes environ, conçoivent et fabriquent des équipements embarqués à bord du satellite ou liés aux activités sol et lanceurs, notamment : le sous-système d'alimentation du satellite les équipements de puissance pour les plateformes et les charges utiles, les bancs de contrôle lanceurs et équipements lanceurs.*

- ***Alcatel Espacio** à Madrid : La société comprend 180 personnes, elle conçoit et fabrique des produits radio-fréquence passifs : Input MultipleXer (IMUX), des équipements de traitement de données numériques Digital Video Broadcasting (système de codage de données), des transpondeurs bande S, des émetteurs en bande KU à partir de 2002.*

2.3 Implication d'Alcatel Space dans les programmes spatiaux

Alcatel a été maître d'œuvre de :

- *5 systèmes complets : Syracuse III, Egnos, WorldSpace, Europe*Star et Rascom.*
- *52 satellites Spacebus : 41 lancés (dont 25 en opérationnels) et 11 en cours.*
- *Plus de 130 charges utiles de télécommunication, incluant 64 Globalstar.*
- *8 satellites d'observation/environnement dont Calipso, Jason 2 et SMOS.*
- *22 charges utiles d'observation/environnement.*
- *11 satellites Météosat : 7 de première et 4 de seconde génération.*

© Alcatel Alenia Space 2005

- *5 satellites scientifiques dont Corot, COGE, Fuego et Herschel Planck*
- *5 charges utiles scientifiques.*

Sans oublier Huygens, à la surface de Titan depuis le 14 janvier 2005.

2.4 Moyens industriels

Alcatel Space dispose de moyens industriels très importants tant pour l'intégration des charges utiles que des satellites, mais aussi en ce qui concerne la production d'équipements électroniques placés à bord des satellites, des lanceurs, mais aussi au sol. Au total, ce sont plus de 15 000 m² de salles blanches qui sont utilisés pour la fabrication des équipements et l'intégration des modules des satellites. Alcatel Space dispose également d'un centre de tests parmi les plus importants d'Europe, notamment pour les tests de simulation spatiale tels que :

- *Compatibilité électromagnétique (EMC).*
- *Thermique.*
- *Acoustique.*
- *Vibration et radiofréquence, calibrés pour tous les niveaux de la chaîne d'assemblage (des équipements aux satellites complets).*

Des moyens de mesure et de test d'antennes du champ proche au champ lointain sont également disponibles. Enfin, le plus grand centre d'optique spatiale en Europe est situé sur le site de Cannes avec 1 000 m² de salle blanche.

2.5 Historique du site de Cannes

***1921** : Etienne Romano, cannois d'origine et grand amateur d'aéronautique, crée l'industrie aéronautique azuréenne avec les Ateliers Romano: 200 avions et hydravions de 11 types vendus en France et en Espagne (chasse, bombardement, transport, multimissions, observation, voltige).*

***1937** : Les ateliers fusionnent avec la SNCASE (Société Nationale de Construction Aéronautique du Sud-Est).*

***1948** : L'établissement, dénommé alors Groupe Technique de Cannes, se consacre à l'étude, la réalisation, et le lancement "d'engins spéciaux" de types variés et faisant appel à une multiplicité de techniques.*

Vue d'avion du site d'Alcatel Space à Cannes

***1956** : La raison sociale change et devient Sud-Est Aviation.*

© Alcatel Alenia Space 2005

***1957** : La fusion avec Sud-Ouest Aviation donne Sud-Aviation.*

***1958-1965** : Un tournant dans la politique de la Défense Nationale donne la priorité à une force de dissuasion. Grâce à ses études antérieures, Cannes reçoit délégation pour l'étude, la réalisation, et les essais d'engins balistiques. A partir de 1965, parallèlement à ses activités militaires, l'établissement de Cannes se lance dans la compétition spatiale en répondant aux appels d'offres du Centre National d'Etudes Spatiales (CNES), de l'Agence Spatiale Européenne (ESA), et enfin à ceux du monde entier.*

***1970** : Création de la société Aérospatiale (fusion de Sud-Aviation, Nord-Aviation et Sereb). Cannes devient maître d'œuvre ou contribue à plus de 200 satellites. Numéro un européen de l'observation et de l'optique spatiale, premières exportations non américaines, plus lointain atterrissage dans le système solaire... Cannes détient la seule installation intégrée de fabrication et d'essais de satellites au monde, hors Californie.*

***2001** : Naissance d'Alcatel Space, de la réunion au sein d'une même société des activités satellites d'Alcatel, d'Aérospatiale, de Thomson-CSF et de Cegelec. Cette nouvelle société, détenue à 51 % par Alcatel et à 49 % par Thomson CSF, est intégrée au sein du groupe Alcatel. Elle est organisée autour de deux secteurs : les systèmes commerciaux par satellite et la gestion de l'ensemble des moyens industriels et de R&D. Ce dernier secteur, baptisé Alcatel Space Industries, abrite l'établissement de Cannes, dans lequel plus de 2050 personnes (3000 avec tous les prestataires), dont plus de 60 % d'ingénieurs et cadres, aident les hommes à mieux communiquer, à observer la Terre et l'univers.*

***2005** : Naissance d'Alcatel Alenia Space. Le 17 juin 2004, Alcatel et Finmeccanica ont signé un accord dans le but de créer, dans le domaine du spatial, deux Joint-Ventures. La première en charge des activités industrielles résulte de la fusion des activités industrielles d'Alcatel Space et d'Alenia Spazio et sera détenue à environ 67% par Alcatel et 33% par Finmeccanica. La seconde en charge des activités de services résulte du rapprochement des activités de services d'Alcatel Space avec Telespazio et sera détenue à environ 67% par Finmeccanica et 33% par Alcatel.*
Après l'accord de l'Union Européenne le 28 avril 2005 sur la fusion des activités spatiales entre Alcatel et Finmeccanica, les travaux préparatoires se poursuivirent pour réussir un démarrage effectif de la société Alcatel Alenia Space (Fusion Alcatel Space - Alenia Spazio) au 1er juillet 2005.

2.6 Les satellites Alcatel Space

Alcatel Space peut fournir un système spatial complet mais aussi des charges utiles, plateformes, équipements satellites et réseaux et des segments sol. Ainsi, la société dispose d'une ligne de produits plateformes, Spacebus 3000 & 4000 pour les satellites géostationnaires et Proteus pour les satellites en orbite basse, qui lui permet de répondre à l'ensemble des besoins de ses clients.

2.6.1 Proteus

Proteus est une famille de plateformes multi-mission à faible coût pour des satellites de jusqu'à 600 kg destinés à être placés en orbite basse pour une durée de vie de 5 ans. C'est à cette famille qu'appartiennent les satellites suivants :

- ***JASON 1** : satellite à vocation océanographique, successeur de Topex-Poseidon, programme en collaboration avec le CNES et la NASA. Ce satellite est la première application de la plate-forme Proteus ; en orbite depuis Décembre 2001, il permet de mesurer le niveau des océans avec une précision de 2 cm².*

© Alcatel Alenia Space 2005

- ***JASON 2*** *: la suite du programme pour une océanographie opérationnelle. Jason-2 conserve, comme Jason-1, la mission de surveillance et de mesure précise des océans. Le satellite Jason-2 comportera également, à titre probatoire, un radar interféromètre: WSOA (Wide Swath Ocean Altimeter) fourni par la NASA permettant d'améliorer la couverture spatiale. WSOA permettra de mesurer des phénomènes méso-échelles tels que les tourbillons à l'intérieur des courants afin d'améliorer les modèles de circulation océanique.*

- ***COROT*** *: destiné à l'étude de la structure interne des étoiles par des techniques de photométrie stellaire de très haute précision, et la recherche de nouvelles planètes.*

- ***CALIPSO*** *: permettra de mieux comprendre le climat de notre planète avec la mesure à l'échelle mondiale de l'influence des aérosols produits par l'activité humaine.*

- ***MEGHA Tropiques*** *: doit permettre une meilleure compréhension des phénomènes climatiques tropicaux grâce à l'étude du cycle de l'eau au niveau des régions tropicales et à la mesure des bilans énergétiques en zones intertropicales.*

- ***SMOS*** *: dédié à la mesure de l'humidité des sols et à la mesure de la salinité des océans.*

2.6.2 Spacebus

La famille de satellites Spacebus est une gamme de satellites de télécommunication geosationnaires, capables de répondre à tous les besoins des clients. Les Spacebus effectuent des missions variées (de bande étroite à large bande) pour des services de télécom fixes, mobiles ou de diffusion. Les Spacebus sont délivrés, au client, selon son souhait, au sol ou en orbite, dans des délais allant de 18 à 30 mois, en fonction de la complexité de la mission.

Voici les principales caractéristiques techniques:

Un satellite de la gamme peut atteindre une masse de 6 tonnes (masse au lancement) et emporter des générateurs solaires délivrant une puissance de 18 kW, dont 15 kW sont disponibles pour la charge utile. Celle-ci peut atteindre une masse de 1100 kg. Les satellites Spacebus ont une durée de vie nominale de 15 ans et sont compatibles avec tous les lanceurs commerciaux.

La gamme Spacebus est basée sur une architecture modulaire, dont la taille peut varier selon les besoins de la mission. Le module de service, construit autour d'un tube central contenant les réservoirs

© Alcatel Alenia Space 2005

d'ergols est intégré et testé indépendamment du module de télécom (la charge utile), ce qui permet un travail en parallèle et raccourcit les délais de production.

Voici les programmes Spacebus pour les années 2004 et 2005 :

- ***WORLDSAT 2** (ex AMC-12), satellite de télécommunications de nouvelle génération (nouvelle plate-forme Spacebus 4000 et nouvelle avionique développée par Alcatel Space) pour l'opérateur américain SES AMERICOM.*

- ***WORLDSAT 3** (ex AMC-23) avec une mission de télécommunications transocéanique pour la nouvelle société WORLDSAT, filiale de SES AMERICOM. (Spacebus 4000).*

- ***APSTAR VI**, satellite de télécommunications de haute capacité (et segment sol associé) pour la Société APT Satellite Holdings Limited , un des principaux opérateurs de la région Asie Pacifique ; ce satellite permettra d'offrir des services multimédia large bande et de télédiffusion numérique mais aussi des services de télécommunications traditionnelles. (Spacebus 4000).*

- ***STAR ONE**, système satellitaire de télécommunications régionales à destination de l'Amérique Latine pour l'opérateur brésilien Star One. (Spacebus 3000).*

- ***KOREASAT V**, pour la société KT Corporation (anciennement Korea Telecom) et l'Agence de Développement pour la Défense (ADD). Ce sera le premier satellite de communication civil et militaire de la République de Corée. Ce contrat a été remporté été 2003. C'est au développement du logiciel de vol de ce satellite que s'attache le stage présenté ici. (Spacebus 4000).*

- ***RASCOM,** premier satellite de télécommunications RASCOM dédié au continent africain. Ce satellite offrira à RascomStar-QAF les capacités requises pour assurer des liaisons fixes voix et données, l'accès à Internet, ainsi que des services de radiodiffusion large bande sur l'ensemble du continent africain. L'empreinte de ce satellite s'étendra d'ailleurs au-delà de l'Afrique, puisqu'elle couvrira également une partie de l'Europe et du Moyen-Orient. (Spacebus 4000).*

- ***HOT BIRD 7A**, pour Eutelsat. La commande de ce nouveau satellite fait suite à la perte du satellite HOT BIRD 7 dans sa phase de lancement. Aux spécifications d'origine d'HOT BIRD 7, ont été ajoutées de nouvelles missions de renfort de capacité et de sécurisation en orbite à la position 13 degrés Est d'Eutelsat, première position mondiale pour le nombre de chaînes de télévision diffusées. (Spacebus 3000).*

- ***SYRACUSE 3A** et **3B**, deux satellites de télécommunication pour la DGA. Le satellite Syracuse IIIA complétera dans le courant 2004 la constellation des satellites mixtes civil/militaire Telecom 2 pour offrir aux forces armées un service très significativement amélioré notamment en termes de débit d'information, de flexibilité d'emploi et de résistances aux contre-mesures. La commande pour le satellite Syracuse IIIB a été notifiée par la DGA fin 2003. (Spacebus 4000).*

- ***GALAXY 17**, nouveau satellite de télécommunication destiné à l'opérateur américain PanAmSat. Il sera le premier satellite européen à intégrer la flotte de PanAmSat, l'un des plus importants fournisseurs mondiaux de services de communications par satellite couvrant les Etats Unis, l'Amérique Latine, l'Afrique, l'Europe, le Moyen Orient et l'Asie. (Spacebus 3000).*

- ***CHINASAT 9**, satellite de télécommunications de nouvelle génération. Ce satellite de télévision directe permettra à ChinaSat d'être la première entreprise institutionnelle chinoise à offrir des services de télédiffusion par satellite en Chine. (Spacebus 4000).*

© Alcatel Alenia Space 2005

2.6.3 Sondes scientifiques

***Huygens** : une sonde interplanétaire lancée en octobre 1997 à bord du véhicule Cassini, analysa l'atmosphère de Titan, le plus gros satellite de Saturne, qu'elle atteint le 14 janvier 2005 (le plus lointain atterrissage jamais tenté et réussi).*

***Herschel** et **Planck** : Successeur de ISO (Infrared Space Observatory lancé en 1995) dont Alcatel Space était maître d'œuvre, Herschel permettra pour la première fois d'étudier, dans le domaine des ondes sub-millimétriques, la formation et l'évolution des galaxies et des étoiles depuis la création de l'univers.*

Ce sera le plus grand télescope spatial jamais lancé avec un miroir primaire de 3,5 mètres. Planck permettra de mieux comprendre l'origine et l'évolution des grandes structures de notre univers juste après le fameux Big Bang. Il mesurera les fluctuations du corps noir cosmologique avec une sensibilité et une résolution sans précédent.

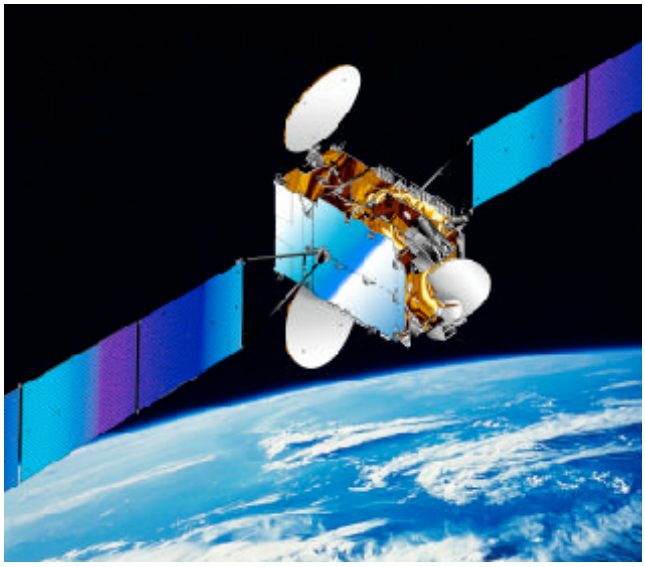
3 Le Spacebus 4000

Intéressons nous à présent au type de satellite Spacebus 4000 et en particulier à son avionique afin de pouvoir mieux situer l'environnement dans lequel s'exécute le logiciel de vol.

3.1 Mission

La gamme Spacebus, des satellites de télécommunication destinés à être placés sur orbite géostationnaire, a été conçue afin de réaliser un grand nombre de missions télécom telles que :

- *Télécommunications militaires en bande X (7 / 8 GHz).*
- *Diffusion numérique audio en bande S (2 / 2.2 GHz).*
- *Services satellitaires en bande C (4 / 6 GHz).*
- *Diffusion et services satellitaires en bande Ku (12 / 14 GHz).*
- *Multi-diffusion haut débit (large bande) en bande Ka (20 / 30 GHz).*



3.2 L'Avionique 4000

L'Avionique 4000 est une avionique moderne entièrement conçue et développée par Alcatel Space pour la gamme de satellite Spacebus 4000, mais dont pourra bénéficier également la plateforme Spacebus 3000. Cette avionique est customisable à tout type de mission et intègre les technologies les plus modernes.

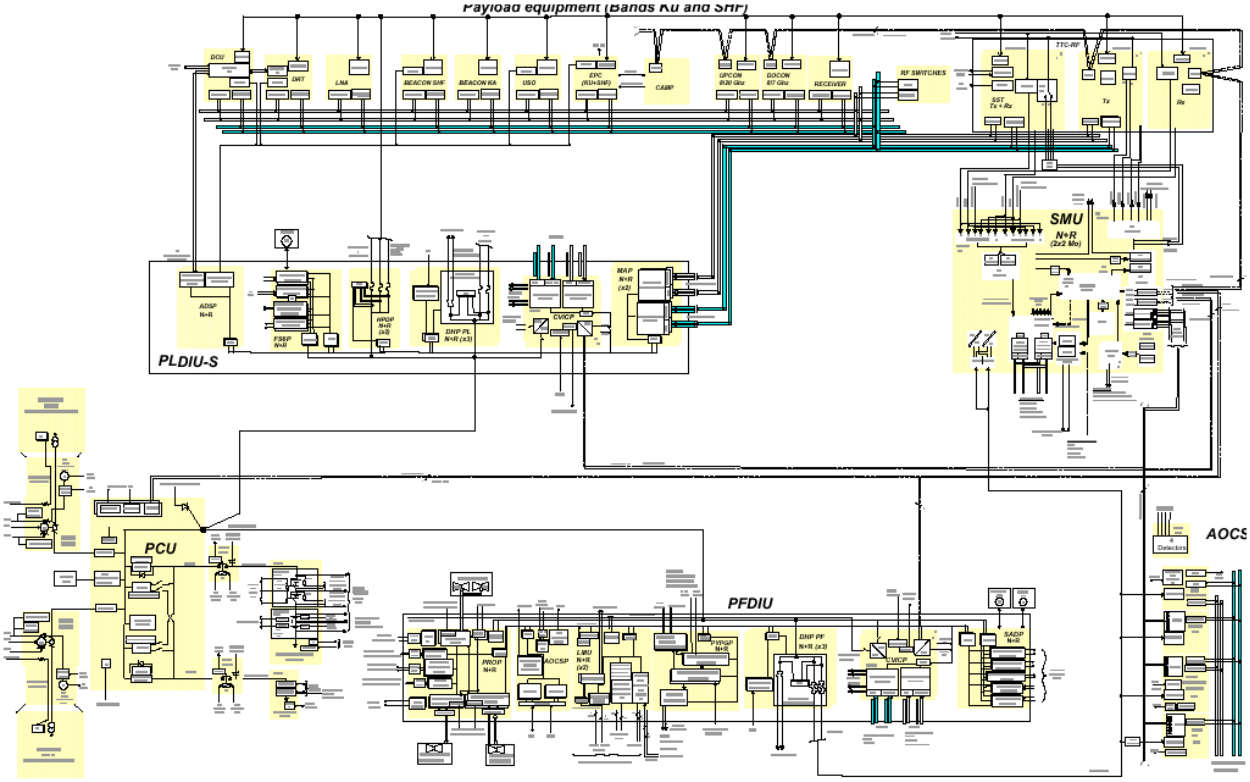


Schéma système de l'avionique 4000

Enumérons les trois chaînes fonctionnelles de la plateforme Spacebus 4000 constituant l'avionique. Deux de ces chaînes seront décrites dans ce rapport plus en détail car étant directement liées au sujet du stage, à savoir l'alimentation électrique implémentée par l'EPS – le système fournissant l'énergie électrique et la gestion bord, matérialisée par le SMU, l'ordinateur de bord. La dernière chaîne fonctionnelle est l'AOCS, le système de contrôle d'attitude et d'orbite et n'est pas concernée par le sujet du stage.

Tous les équipements, les cartes électroniques constituant l'avionique sont redondés.

3.2.1 EPS – Electrical Power Supply

L'EPS, le sous-système fournissant la puissance électrique, est composé de différents éléments :

- *Le Bearing And Power Transfer Assembly (BAPTA) ou Solar Arrays Drive Motor (SADM) qui délivre la puissance électrique via des contacteurs tournants au PCU – power conditioning unit.*
- *Une ou deux batteries Lithium Ion, (selon la configuration) chacune composée de deux modules de batteries, eux-même constitués de cellules assemblées en série.*
- *Le PCU – Power Conditioning Unit ou unité de conditionnement de puissance, qui reçoit la puissance soit par les panneaux solaires soit par la batterie durant le lancement ou pendant les éclipses dont la fonction est la régulation – maintenir le bus d'alimentation à la tension de 100 Volts .*

© Alcatel Alenia Space 2005

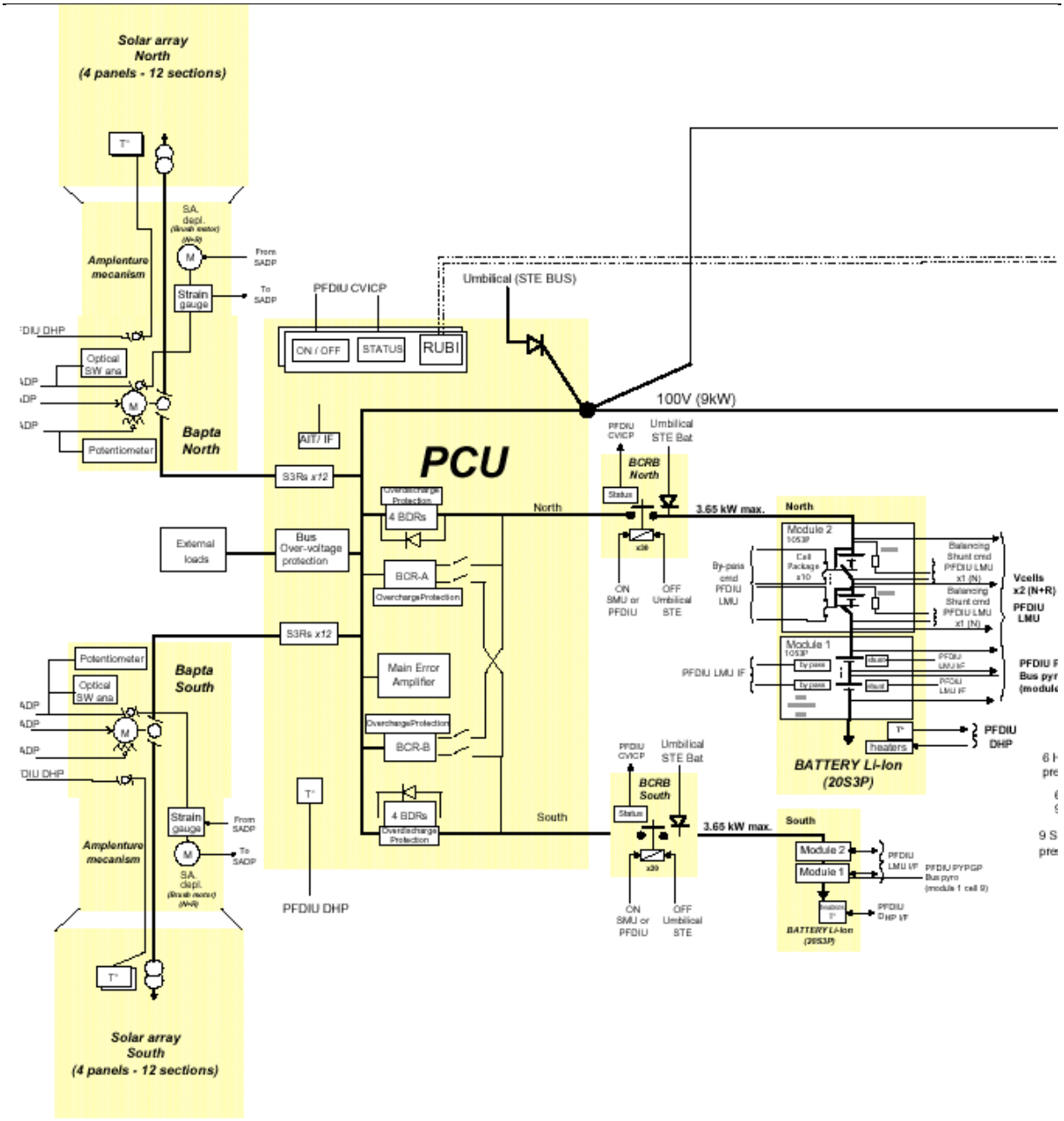


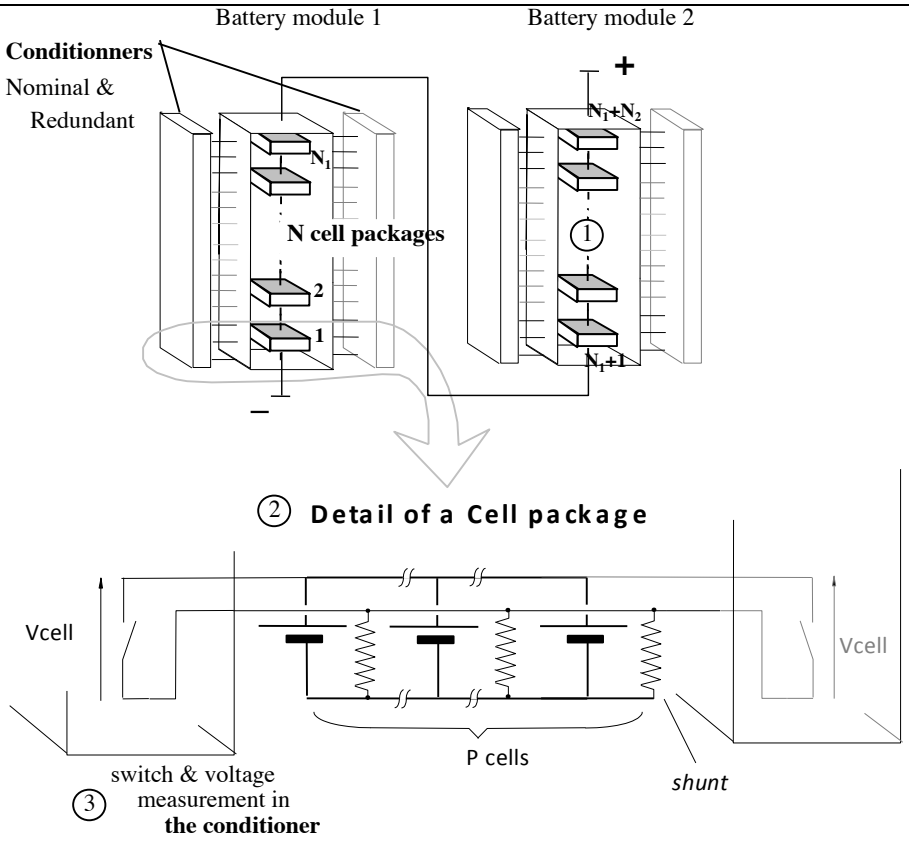
Schéma du système de puissance électrique

3.2.1.1 Batteries

Voici la description d'une batterie et de son électronique associée :
Ainsi qu'écrit précédemment, une batterie est composée de deux modules et le nombre de batteries emportées par le satellite est de un ou deux.

Un module de batterie est un assemblage de 9 à 12 groupes de cellule (appelés simplement cellules par commodité) connectés en série afin de fournir la tension nécessaire. Un groupe de cellules est lui-même composé d'un certain nombre de cellules en parallèle fournissant la capacité voulue.

© Alcatel Alenia Space 2005



Deux cartes électroniques (identiques, une nominale et une autre en redondance froide) appelées LMU – Lithium-Ion Management Unit, sont associées à chaque batterie. Cette carte LMU permet l'acquisition des valeurs de tensions pour chaque cellule, ainsi que l'activation de shunts utilisés pour équilibrer l'état de charge des groupes de cellules. Des fonctions de switch (pour « bypasser » – contourner une cellule) sont également implémentées sur la carte. Ces fonctions servent à isoler les cellules défectueuses.

La carte LMU peut fournir, pour télémétrie, les données suivantes : les tensions des cellules et les tensions des groupes de 6 cellules. De plus, l'état des switches de bypass peut également être descendu en TM.

En cas de dépassement de la tension maximale autorisée, un signal OVP – OverVoltage Protection – est généré afin de reconfigurer l'EPS et recommencer un contrôle correct de la charge.

3.2.1.2 PCU – Power Conditioning Unit

La fonction principale du PCU, l'unité de conditionnement de puissance, est de réguler le bus d'alimentation à la tension de 100 V. Pendant les périodes d'éclairement solaire, la puissance électrique est délivrée par les panneaux solaires à travers une série de shunts. Lorsque la puissance fournie par les panneaux solaires est insuffisante ou nulle (en éclipse) les batteries fournissent les équipements du satellites en énergie au travers de plusieurs BDRs – Battery Discharge Regulators. Le PCU est également responsable de la recharge des batteries et utilise pour cela le BCR – Battery Charge Regulator – en redondance froide ainsi qu'un séquenceur.

Le séquenceur peut fonctionner dans différents modes. Il permet la charge de la batterie 1, de la batterie 2, des deux batteries sumultanément, ou enfin alternativement, avec une période d'une dizaine de minutes.

© Alcatel Alenia Space 2005

La régulation du bus de puissance est basée sur l'observation des variations du courant électrique sur le bus.

Lorsque la puissance délivrée par les panneaux solaires pour alimenter à la fois le bus et les batteries est suffisante voire excédentaire, la régulation est effectuée par les shunts (mode S3R) qui connectent ou déconnectent des section s de panneaux solaires. Puis, lorsque la puissance diminue, le courant de charge est réduit, tous les shunts sont connectés au bus afin de collecter la puissance de l'intégralité des panneaux solaires et c'est le BCR qui prend le relais pour assurer la régulation – le maintien des 100 V (mode BCR). Finalement, lorsque le satellite n'est plus en éclairement solaire, les BDR gèrent la décharge des batteries (mode BDR).

3.2.2 SMU – Satellite Management Unit

Le SMU est l'ordinateur de bord du satellite. Il contient le module processeur (PM – Processor Module), le module central de reconfiguration (CRM – Central Reconfiguration Module), la mémoire de contexte (SGM – Safe Guarded Memory), les interfaces TM/TC, les interfaces avec les bus OBDH et 1553B. L'avionique 4000 comporte deux SMU, le nominal et le redondant.

Les bus de communication OBDH et 1553B sont utilisés au travers des services logiciels HDSW, qui seront décrits dans la prochaine section. Ces bus permettent de dialoguer avec les équipements matériels du satellite afin de leur faire effectuer des actions ou pour obtenir des informations sur leur état.

Le CRM, module de reconfiguration central, assure la gestion des modes du calculateur (SMU). Il est également responsable de la détection et de la validation des alarmes. Le CRM génère de façon autonome les commandes de reconfiguration sur détection d'alarme ou sur requête logicielle. Le CRM est commandé par la fonction logicielle FDIR (détection des erreurs et reconfigurations).

La mémoire de contexte (SGM) est utilisée par le logiciel de bord afin d'enregistrer le contexte logiciel, incluant le vecteur d'état du système de contrôle d'attitude, la séquence d'évènements déroulé, l'historique des erreurs, la configuration du satellite, etc. Le but est de pouvoir sauvegarder les données afin de les réutiliser après une panne importante nécessitant une reconfiguration (redémarrage) du calculateur. Il y a en réalité deux modules SGM (SGM A et SGM B), chacun composé de deux parties de respectivement 128 kb et 512 kb.

© Alcatel Alenia Space 2005

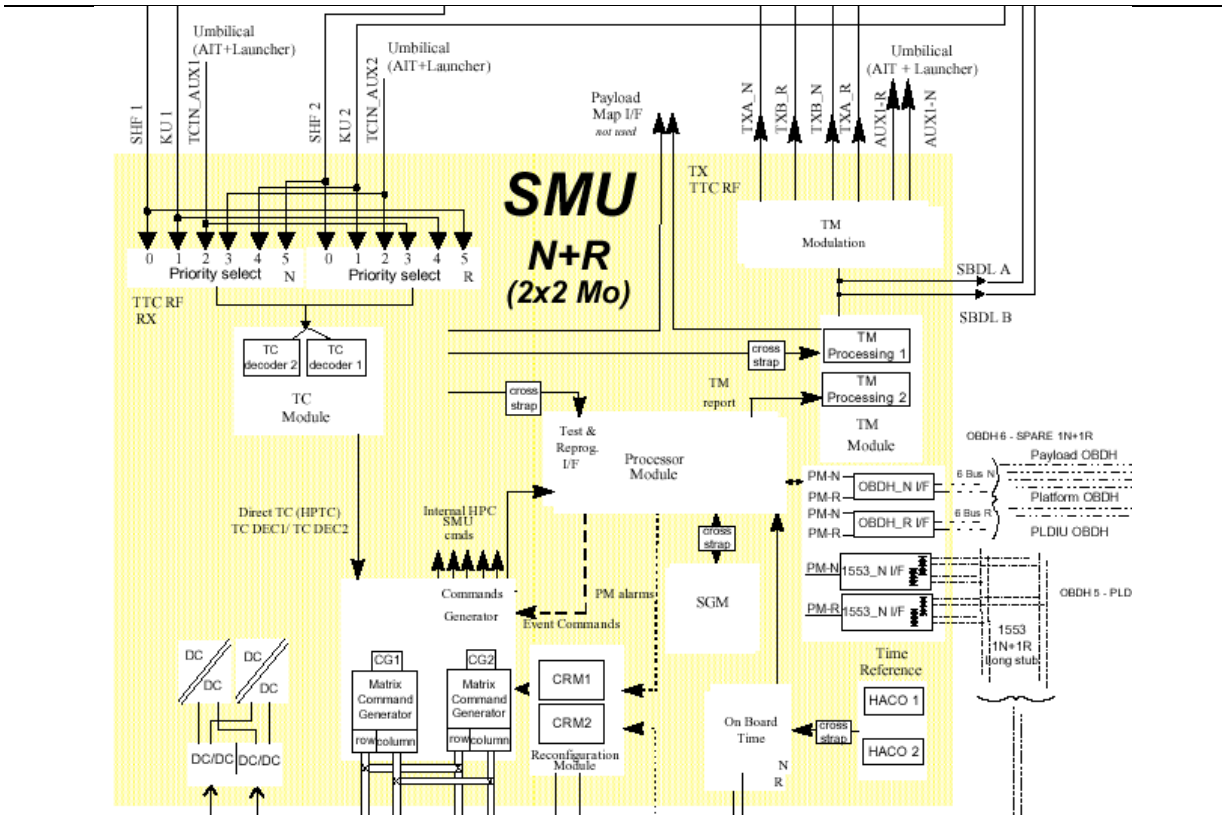


Schéma du calculateur de l'avionique 4000

Tout au long de la mission, le logiciel doit maintenir la consistance des données de la SGM, c'est-à-dire que les paramètres logiciels doivent être mis à jour lorsque leur valeur change dans la mémoire vive.

3.2.3 AOCS – Attitude and Orbit Control Subsystem

La particularité du système de contrôle d'attitude et d'orbite est qu'il utilise des roues à réaction sans moment cinétique embarqué (RW – Reaction Wheel, 3 Nominales + 1 Redondante) ainsi que des senseurs stellaires (Star Tracker, 1N+1R). L'avionique 4000 est la première avionique de satellite sur orbite GEO à intégrer des senseurs stellaires. Il comporte cependant également des équipements plus classique tels que des senseurs solaires (CSS – Coarse Sun Sensor, 1N+1R), des senseurs terre (IRES – InfraRed Earth Sensor, 1N+1R) et des gyromètres (GYRO, 1N+1R)

3.3 Le logiciel de vol : OBSW

Le logiciel de vol (OBSW – On Board SoftWare) est le programme informatique exécuté dans l'ordinateur de bord du satellite et dont la tâche est la gestion des systèmes tels le système de contrôle d'attitude et d'orbite, le système de puissance (comprenant les batteries), le système de régulation thermique, le système de communications assurant le lien bord-sol, etc. Ce logiciel de vol doit aussi gérer les pannes et les reconfigurations rendues nécessaires afin de neutraliser les conséquences des défaillances possibles de certains équipements. Pour réaliser cela, chaque fonction est associée à une ou plusieurs tâches logicielles.

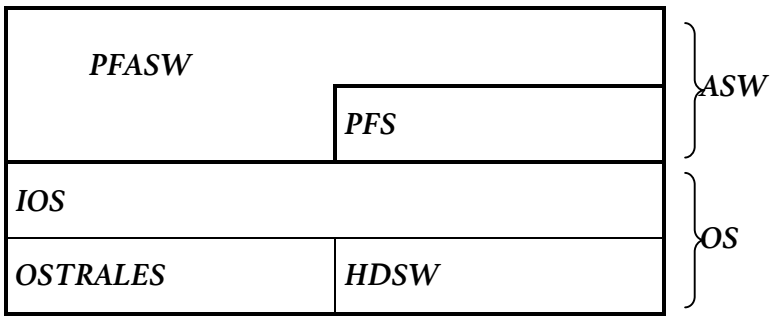
© Alcatel Alenia Space 2005

Le logiciel de vol, qui sur le Spacebus 4000 possède une grande responsabilité et donc criticité, est l'un des points sensibles de la plateforme. Il fait dès lors l'objet d'un soin particulier dans son développement et sa validation afin de le rendre aussi sûr et fiable que possible.

Ce logiciel est codé dans le langage Ada (décrit plus en détails dans une autre partie du rapport), un langage développé spécifiquement pour ce type d'application. Le logiciel de vol est composé de plusieurs parties, qui s'assemblent en couches « horizontales » et « verticales ».

3.3.1 Architecture

Le logiciel de vol est structuré en plusieurs couches, interagissant les unes avec les autres : La couche inférieure est la couche OS – Operating System – le système d'exploitation. La couche supérieure, ASW – Application SoftWare – est la couche d'applicatifs, représentant tout ce qui fait "l'intelligence" du satellite. La couche intermédiaire IOS forme une interface entre l'OS et les logiciels applicatifs, facilitant aux logiciels de haut niveau l'accès aux ressources de bas niveau.



Couches logicielles de l'OBSW

3.3.2 Couche OS – Operating System

Le système d'exploitation (OS) fournit toutes les interfaces haut niveau avec le hardware et est responsable de l'ordonnancement des tâches.
L'OS est subdivisé en trois entités logicielles : OSTRALES, HDSW et IOS.

3.3.2.1 OSTRALES

OSTRALES – Operating System Temps Réel Adapté aux Logiciels Embarqués Spatiaux – est l'ordonnanceur qui gère l'accès à la ressource partagée qu'est le processeur. Il active et désactive les tâches de plus haut niveau exécutées de manière cyclique ou asynchrone (acyclique).

OSTRALES a été développé par Alcatel Space, et est écrit dans sa quasi totalité en Ada83.

D'un point de vue technique, ce dernier :

-Répond seulement aux besoins connus des logiciels de bord (réduction de la complexité, suppression du code mort),
-Offre un ensemble de fonctionnalités à une application temps réel :
gestion des processus,
gestion du temps,
gestion des interruptions,
gestion de la synchronisation,
gestion du partage de ressources.

© Alcatel Alenia Space 2005

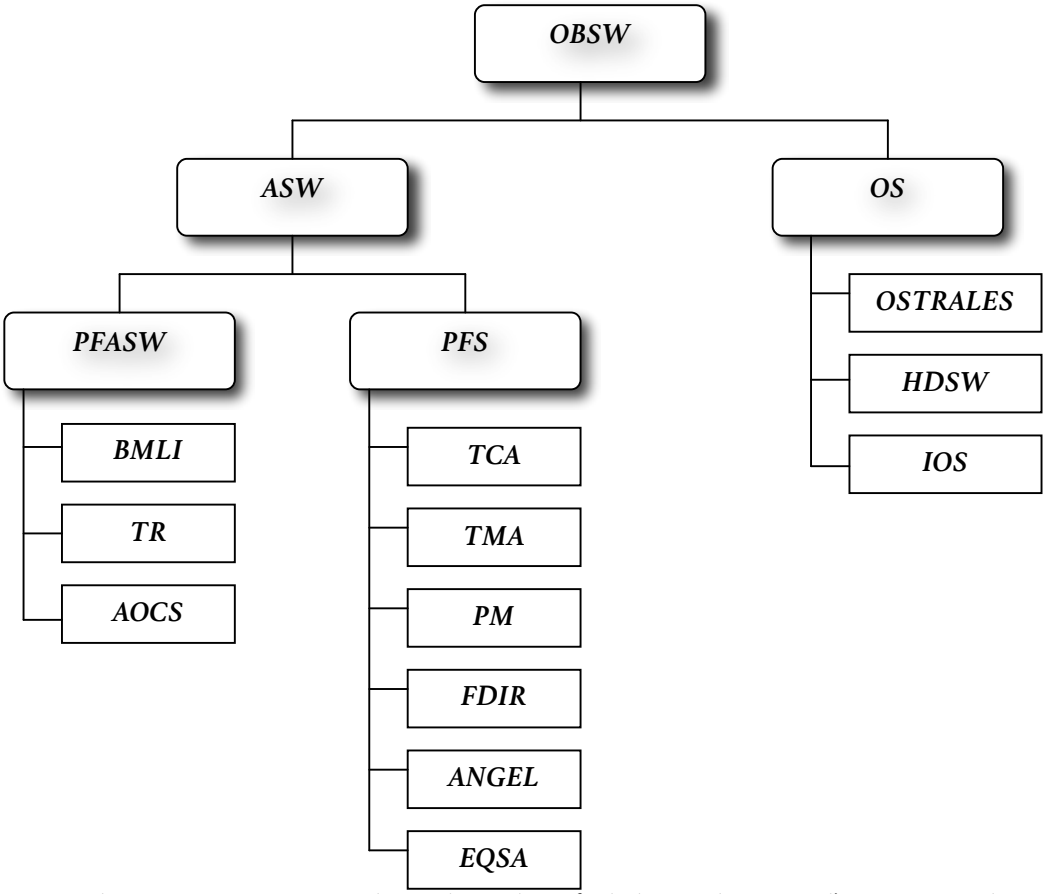
L'ordonnancement des tâches est préemptif (une tâche peut être momentanément stoppée, mise de côté et redémarrée ultérieurement losrqu'une autre tâche, de plus haute priorité, doit être exécutée) et sa politique est d'être à tout moment en train d'exécuter la tâche la plus urgente, celle dont l'échéance est la plus proche.

Hiérarchie des composants logiciels

Les priorités sont figées, c'est-à-dire qu'elles sont déterminées par avance, durant la phase de spécification, de design. Il ne s'agit pas ici d'un ordonnancement à priorités variables (recalculées dynamiquement en fonction du temps restant pour arriver à l'échéance).

3.3.2.2 HDSW – Hardware Dependent SoftWare

HDSW, logiciel dépendant du hardware, est l'interface entre les ressources matérielles et la partie logicielle. Les principales ressources matérielles sont les bus de communication : le bus OBDH et le bus



1553B. Le bus OBDH permet aux logiciels applicatifs de la couche ASW d'envoyer une liste de tous types d'interrogations, acquisitions et commandes vers les équipements du satellite. Le bus 1553 permet l'envoi par l'ASW d'une simple commande ou d'une liste de simples commandes.

Trois points de connexion sont directement reliés au SMU afin qu'il puisse déterminer (par vote majoritaire) si le satellite est encore lié au lanceur. Ce service est fourni par HDSW.

C'est le service HDSW qui fournit l'heure (OBT – On Board Time) codée sur 48 bits avec une résolution de 1 ms.

© Alcatel Alenia Space 2005

Deux décodeurs de TC reçoivent les télécommandes en provenance du sol et les délivrent au module processeur. Le protocole est le suivant : sur réception d'un paquet de TC dans le décodeur, une interruption est levée et le logiciel bord invoque le service HDSW qui copie le paquet de TC dans la RAM.

3.3.2.3 IOS – Input Output Services

IOS est une couche de fonctions de services gérant les entrées – sorties au niveau du système d'exploitation.

3.3.3 Couche ASW – Application SoftWare

Enfin la couche la plus haute, nommée ASW, est la partie du logiciel effectuant les opérations les plus complexes. Il s'agit d'un logiciel haut niveau traduisant le fonctionnement du satellite.

La couche ASW est composée de deux groupes : le groupe de logiciels de services (PFS) et le groupe d'applications système (PFASW).

3.3.3.1 PFS – PlatForm Services

Ce groupe contient toutes les fonctions OBSW fournissant des services. Ces fonctions sont génériques, c'est-à-dire qu'elles ne dépendent pas de la mission du satellite, ni de sa configuration.

Voici la description de ces fonctions :

3.3.3.1.1 TCA – TeleCommand Application

Cette application gère tout ce qui concerne les télécommandes. Elle reçoit le flux de télécommandes émises par le sol et après vérification de la validité et de la syntaxe les redistribue vers les applications correspondant au type de télécommande ou alors directement vers les bus OBDH ou 1553 lorsqu'une commande de bus de données est reçue.

3.3.3.1.2 TMA – TeleMetry Application

L'application TMA se charge du flux de télémétries à renvoyer au sol. Elle construit les champs de données des paquets de TM, en tenant compte du format utilisé.

Cette application est également responsable de l'acquisition des données demandées par le sol.

3.3.3.1.3 PM – Platform Management

Le « platform management » est responsable des manipulations dépendantes de la plateforme, excepté la gestion du bus qui est du ressort d'autres applications.

Cette application est également en charge de la gestion de différents paramètres du calculateur, tel que la mémoire de contexte SGM, ainsi que la détermination du mode et de la phase du satellite. Cette fonction est responsable du basculement d'un mode à un autre selon le type de panne qui peut se produire.

3.3.3.1.4 FDIR – Failure Detection, Isolation and Recovery

La détection d'erreurs, leur isolation et le retour à un état de fonctionnement est la tâche de la FDIR. Il existe 4 niveaux d'erreurs, selon la gravité de la défaillance. Chaque erreur conduit à l'exécution d'une séquence d'évènements visant à reconfigurer les systèmes pour passiver les pannes.

© Alcatel Alenia Space 2005

D'autre part, cette application est en charge de la gestion de l'historique d'erreurs (FHB – Failure History Buffer) ainsi que du rafraîchissement périodique de la mémoire de contexte SGM.

3.3.3.1.5 ANGEL – Avionics New Generation of ELectronics

ANGEL est une application de services facilitant le dialogue et la commande des équipements via le bus de communication OBDH.

Plus précisément, ANGEL est chargé de gérer le contenu en émission et réception des trames de données circulant sur le bus OBDH mais aussi le timing de celles-ci d'où la notion de temps réel. De plus, ANGEL détecte les erreurs sur le bus de communication et en notifie la fonction FDIR. Par ailleurs, cette couche s'occupe, dans les actions unitaires, de gérer les équipements : contrôle des différentes cartes, mise en œuvre de fonctionnalités (capteurs, actionneurs), initialisation de registres internes.

Mais elle peut également réaliser des fonctions complexes telles que le pivotement de panneaux solaires dans le but de conserver un ensoleillement optimal ou l'alimentation des cartes en les connectant au bus d'alimentation. Cette gestion de fonctions complexes est assez récente. En effet, ANGEL, contrairement aux autres briques de la plate-forme SB4000, a été entièrement repensée. Désormais, chaque carte électronique analysée sera décomposée en fonctions, et chaque fonction aura son propre service Angel exécutant une liste d'actions de manière autonome.

3.3.3.1.6 EQSA – Equipment and AOCS Sensor / Actuator

L'application EQSA est chargée de l'envoi des commandes et de la réception des réponses sur le bus 1553B. De la même façon qu'ANGEL, l'application détecte les erreurs sur le bus et en fait part à la FDIR. De plus, EQSA fournit des services relatifs aux interfaces des senseurs / actuateurs du système de contrôle d'attitude.

3.3.3.1.7 SWIN – SoftWare Initialisation

SWIN est la fonction responsable de l'initialisation logicielle.

3.3.3.2 PFASW – PlatForm Application SoftWare

Ce groupe contient toutes les fonctions de niveau supérieur. Ces applications reposent sur les fonctions de services décrites précédemment et sont, à quelques exceptions près, indépendantes les unes des autres (par exemple les températures sont acquises par l'application de régulation thermique (TR) puis utilisées par celle gèrant la batterie (BMLI)). Ces applications dépendent de la mission du satellite. Par exemple, KoreaSat V est équipé de batteries Lithium Ion alors que d'autres satellites SpaceBus 4000 utilisent des batteries au NiH2. Dans ce cas, BMLI est remplacé par BMNI.

3.3.3.2.1 BMLI – Lithium-Ion Battery Management

Cette application est responsable de la gestion des batteries, de leur charge, décharge et de la surveillance de tous les paramètres. BMLI est également en charge de la détection du début et de la fin des éclipses.

L'application BMLI est celle qui nous concerne au premier plan, le stage étant rattaché au développement de cette partie du logiciel de vol.

L'ensemble des fichiers source Ada constituant BMLI, commentaires compris, représentent environ 40'000 lignes de code.

© Alcatel Alenia Space 2005

La gestion de la batterie est effectuée par deux fonctions logicielles de haut niveau : 1) la gestion de la recharge et 2) la surveillance et la détection d'anomalies (monitoring and failure detection). Chacune des deux fonctions se divise ensuite en fonctions plus spécialisées, décrites ci-après :

Gestion de la recharge

1) *La gestion des modes, qui détermine si les batteries sont en charge, en décharge ou en pause.*
2) *L'équilibrage des blocks, qui permet de maintenir un équilibre dans l'état de charge des blocks de cellules constituant les batteries.*
3) *Le pilotage de la charge, qui suit et contrôle le bon déroulement de la recharge des batteries.*
4) *L'acquisition des tensions, qui fournit les voltages des cellules et qui indique s'ils sont corrects ou erronés.*
5) *La fonction d'isolation (bypass), qui permet de se débarrasser d'une cellule qui a été déclarée défectueuse.*

Surveillance et détection d'anomalies

1) *La surveillance des interfaces TM/TC, BCR & PCU, qui permettra une reconfiguration des équipements.*
2) *La surveillance des cartes LMU, afin de détecter une erreur provenant de celles-ci.*
3) *La corrélation de voltages, pour reconfigurer l'unité de conditionnement en cas d'erreur.*
4) *La surveillance de décharge excessive, dont le but est de déceler un problème au niveau de l'état de charge des batteries.*
5) *La surveillance des cellules, afin de détecter les cellules défectueuses.*
6) *La protection en surtension des batteries, afin de protéger les batteries contre une éventuelle surtension.*

3.3.3.2.2 TR – Thermal Regulation

Cette fonction se charge de la régulation active de la température de la plateforme et de la charge utile.

3.3.3.2.3 AOCS – Attitude and Orbit Control Subsystem

Le contrôle de l'attitude et de l'orbite du satellite est assuré, du point de vue logiciel, par l'application AOCS, quelle que soit la phase dans laquelle le satellite se trouve (excepté le lancement). L'application gère les équipement affectés à ce contrôle, les met en fonction ou les déconnecte. Le logiciel effectue également une surveillance afin de déceler une éventuelle défaillance. Dans un tel cas, l'application FDIR en est avisée et procède aux reconfigurations.

Pour mener à bien sa tâche, la fonction AOCS acquiert les données en provenance des senseurs (senseurs stellaires, senseurs terrestres infrarouges, etc) par l'interface senseurs / actuateurs via l'application EQSA, effectue les calculs dont elle a la charge puis envoie ses commandes aux actuateurs (roues à réaction, etc) par la même interface en utilisant à nouveau les services offerts par EQSA.

3.3.4 Contraintes

Le logiciel de vol doit naturellement se soumettre à un certain nombre de contraintes établies à la conception. Voici les principales contraintes :

1) *Le logiciel de vol doit être codé en Ada.*
2) *Le mécanisme des tâches (task) possibles avec le langage Ada ne doit pas être utilisé.*
3) *L'allocation dynamique de mémoire ne doit pas être utilisée.*
4) *La mémoire vive (RAM), d'une taille de 4 Mb est divisée en*
   - 3.5 Mb pour les instructions, les opérandes (données et constantes).
   - 0.5 Mb pour les patchs et les « free functions » (fonctions initialement vides, pouvant être

© Alcatel Alenia Space 2005

utilisées pour par exemple réaliser une modification de code de dernière minute, sans retoucher l'OBSW).

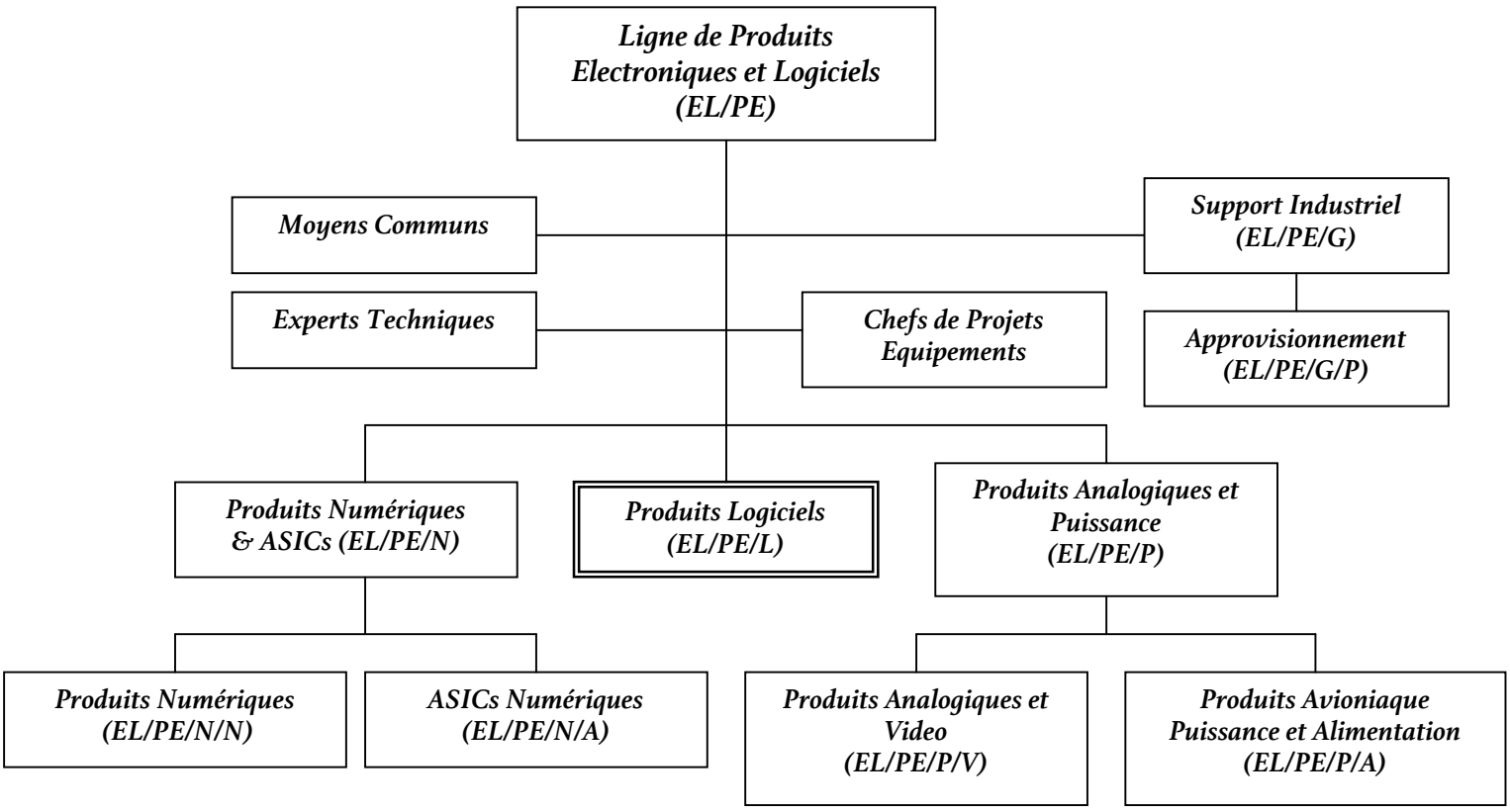
5) *Les instructions, les constantes et les données de l'OBSW doivent utiliser moins de 4Mb dans la RAM avec 30 % de marge (pour les patchs et free functions).*
6) *L'image en PROM du code et des constantes de l'ASW doit avoir une taille inférieure à 2 Mb avec 30 % de marge.*
7) *L'ASW doit utiliser moins de 60 % de la charge de calcul du processeur.*

© Alcatel Alenia Space 2005

4 Le département EL/PE

Le département d'accueil du stage est le département Ligne de Produits Electroniques et Logiciels (EL/PE) d'Alcatel Space, situé dans l'établissement de Cannes. Il fait partie de l'Industrial Unit electronique.

4.1 Responsabilités, structure, organigramme



Le Département Ligne de Produits Electroniques et Logiciels (EL/PE) est une entité opérationnelle de la Direction de l'Industrial Unit Electronique (IUEL) d'Alcatel Space. EL/PE est, en terme de produits couverts, présente sur les sites de Toulouse et de Cannes. Il couvre l'ensemble des Produits Electroniques et Logiciels à l'exception des Produits Hyperfréquence et Antennes de l'IUEL. Il coordonne de façon directe l'ensemble des produits électroniques et logiciels actuellement développés et réalisés sur le site de Toulouse et de Cannes, et fonctionnellement ceux sous-traités dans les filiales.

Le département a pour mission de réaliser :
Les Produits et Equipements électroniques ainsi que les logiciels embarqués tant sur les satellites sous responsabilité Alcatel Space que sur des satellites réalisés par d'autres

© Alcatel Alenia Space 2005

Maîtres d'oeuvre :
- *Equipements numériques à dominante Traitement du Signal.*
- *Equipements numériques à dominante Traitement de Données.*
- *ASICs et/ou FPGA numériques.*
- *Logiciels de vol de niveau application satellite ou enfouis au sein des équipements livrés par EL/PE.*
- *L'ensemble des équipements Avionique.*
- *Les équipements de Puissance et les alimentations intégrées dans les équipements et/ou livrés à des clients externes.*
- *Les équipements électroniques Vidéo pour instruments des CU observations et sciences.*
- *ASICs analogiques full custom.*

En spécifiant les moyens de test et de validation nécessaires à toutes ces opérations.

Le département prend en charge le développement des logiciels de vol embarqués sur les satellites, autant dans la charge utile que sur la plateforme elle-même.
Outre la création des logiciels, le département effectue la validation de ceux-ci, c'est-à-dire la vérification du fonctionnement correct du programme.

Le LV assure le suivi des opérations lorsqu'il fait appel à des tiers (sous-traitants) pour lui prêter main forte dans le développement.

Le département possède donc un rôle majeur dans la mise en œuvre de satellites :
en amont, il propose des actions de recherche et développement, il se conforme aux contraintes de délais et coût imposées, il est garant de la fonctionnalité (logicielle) du satellite et il participe aux développements des évolutions techniques.

4.2 Le service Produits Logiciel : EL/PE/L

C'est le service d'accueil du stage. Le service Produits Logiciels s'occupe de la réalisation des logiciels de vol embarqués sur la plate-forme des satellites, ou sur les charges utiles et équipements. Il intervient et suit à ce titre toutes les étapes de la création du logiciel : conception, définition, réalisation et tests.

Pour cela il s'appuie sur une spécification des besoins formulés par le client et pour réaliser les tests qui lui incombent, il a recours à un moyen de test dédié : le banc de validation logiciel (BVL), dont il a la charge de la spécification, de la mise en place, et de la maintenance.

Outre cette activité principale, il s'occupe activement de la gestion des logiciels et de leurs méthodes de production. En effet, un effort de normalisation et de standardisation est fourni pour chacune des étapes de la création d'un logiciel (conception, développement, validation, stockage, mise en configuration). Et ce dans un but de cohérence et de faciliter la compréhension, la récupération et réutilisation d'existants.

Deux groupes se distinguent encore au sein de ce service :
- Groupe Logiciel Satellite. Ce groupe s'occupe des logiciels embarqués sur la plate-forme de satellites, c'est à dire ceux maintenant le satellite en vol, et permettant de le manipuler. Lorsque le satellite est lancé et est autonome, la station sol ne peut plus communiquer avec lui que par télécommandes (TC), ou par télémesures (TM). Il faut donc que le satellite soit capable de comprendre et traiter ces informations de manière logicielle. Les logiciels embarqués sont donc paramétrables par TC, modifiables. Ces logiciels sont dépendants de l'ingénierie système.

- Cellule Méthodes. C'est cette entité qui se charge principalement de standardiser le processus de développement et d'harmoniser les pratiques.

© Alcatel Alenia Space 2005

5 Le processus de développement logiciel

5.1 Le cycle en V

5.1.1 Principe

Le développement et la mise au point d'un logiciel contrôlant un satellite de plusieurs tonnes orbitant dans un milieu hostile à des dizaines de milliers de kilomètres de la terre nécessite une certaine organisation. Un des modes d'organisation les plus efficaces permettant de mener à bien un tel projet est le fameux cycle de développement en V, largement répandu dans les processus industriels.

Le principe d'application du cycle en V est de commencer par définir les spécifications de haut niveau d'abstraction, puis par étapes d'ajouter des détails tout en découpant la description de chaque entité, tâche ou fonction en plusieurs unités jusqu'à atteindre un niveau de spécification élémentaire. Cette partie est symbolisée par la partie gauche du « V », descendant vers un bas niveau d'abstraction.

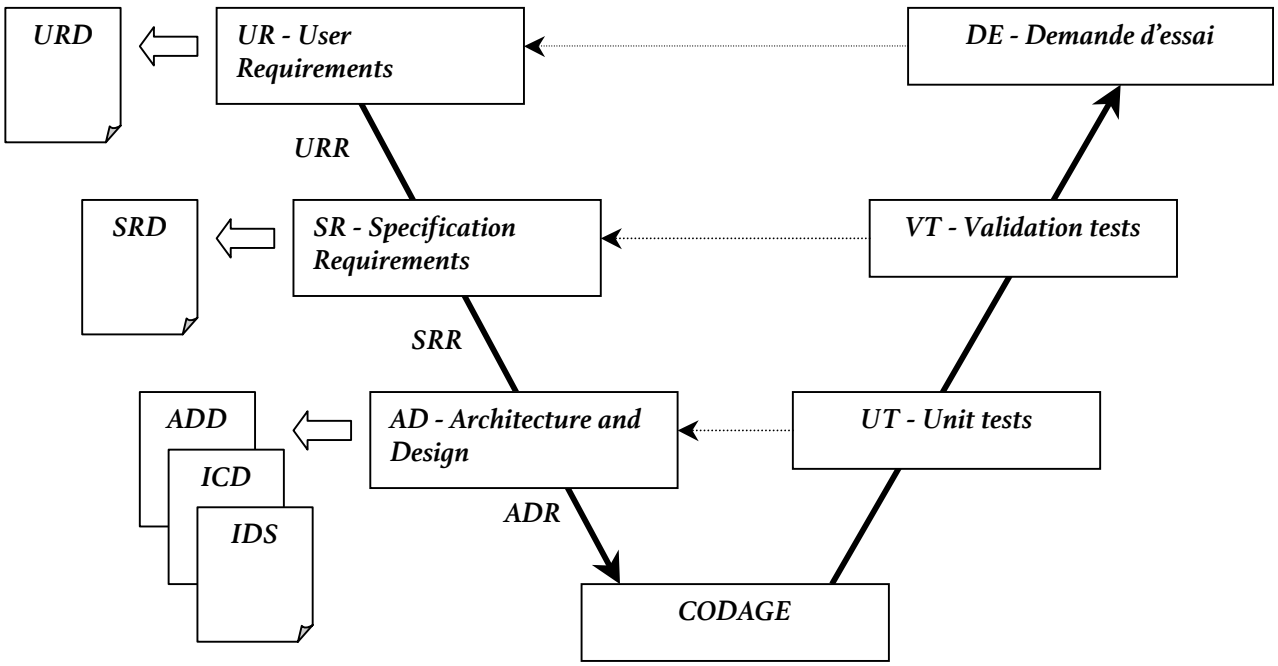
Tout au bas du V, lorsque tous les composants du produit à développer sont parfaitement décrits, la réalisation est effectuée. Dans notre cas, il s'agit du codage des applications, c'est-à-dire de la rédaction du corps des fonctions et procédures Ada.

La partie remontante du V consiste à valider, tester, vérifier, bref à s'assurer du fonctionnement correct et correspondant aux spécifications édictées dans la première phase. En remontant par étapes, les mêmes étapes que dans la partie descendante, on commence par tester chaque unité séparément (étape des test unitaires) puis on réassemble les morceaux constituant le produit et on effectue les tests à un niveau d'abstraction plus haut (tests d'intégration).

Le travail de validation nécessite généralement passablement plus de temps et d'attention que le travail de « fabrication » (de codage, dans notre cas). Cela s'observe facilement en comparant par exemple le nombre de lignes de code constituant le logiciel au nombre de lignes des procédures de test. Le rapport dépasse un pour deux, et ceci uniquement pour les tests unitaires.

© Alcatel Alenia Space 2005

Les pro chaines sections présentent en plus amples détails chacune des trois parties principales du cycle en V pour le développement logiciel : la conception et spécification, le codage et enfin la validation.



5.1.2 Conception et spécification

5.1.2.1 Phase UR – Spécification client

La phase de spécification client est une phase de définition des besoins de l'utilisateur (du satellite). Elle vise à produire la définition du cahier des charges et des interfaces du logiciel de vol OBSW avec les composants physiques du satellite sous la forme d'un document de référence appelé l'URD – User Requirements Document. C'est l'équipe système qui se charge de cette tâche.

Cette phase est validée à l'issue d'une revue : l'URR – User Requirements Review.

5.1.2.2 Phase SR – Spécification logicielle

La spécification logicielle correspond à la phase d'analyse du projet. Le document d'entrée de cette phase est l'URD, le document de sortie de la phase précédente. Au terme des spécifications du logiciel de vol, le document de sortie est la SRD – Software Requirements Document – édictant toutes les exigences que doit respecter le logiciel. La SRR – Specification Requirement Review – est la réunion permettant de valider cette phase.

Cette étape est très importante pour la validation logicielle, car le SRD servira de support pour l'écriture des tests. En effet, il décrit rigoureusement (et de manière détaillée) les fonctionnalités du logiciel ainsi que toutes les exigences (spécifiées dans les URD).

Dès cette phase de spécifications, les ingénieurs chargés de la validation commenceront à réfléchir sur la façon de vérifier que le logiciel embarqué répond à toutes les exigences du ou des SRD. C'est pendant cette phase que le plan de validation est défini : il faut que chaque exigence figurant dans les SRD soit couverte par au moins un test. De plus le logiciel doit être testé en « boite noire », c'est à dire qu'il ne peut

© Alcatel Alenia Space 2005

être manipulé que par des TM / TC car le satellite (et donc le logiciel) est appelé à devenir autonome et ne pourra communiquer que par une succession de TC et de TM.

La deuxième étape est la conception ; elle a pour objectif de définir l'architecture globale et les interfaces entre le logiciel et les autres composants du système (adresses des équipements, format des échanges de donnée). La difficulté de cette étape réside dans la factorisation des problèmes et dans leur résolution. Les problèmes communs peuvent être l'organisation des acquisitions et des commandes, les cycles de régulation définissant le temps d'occupation du microprocesseur, l'organisation des données (tableaux, fichiers), leur représentation (entier, réel simple ou double précision).

5.1.2.3 Phase AD – Spécification détaillée

La phase AD – Architectural Design – vise à définir la structure du logiciel, en partant de la SRD. Cette phase produit un document appelé ADD – Architectural Design Document. L'ADD spécifie notamment toutes les différentes tâches (au sens informatique du terme), les tâches cycliques (ou périodiques) comme les tâches asynchrones (ou apériodiques). L'ADD fixe leur périodes (pour les tâches cycliques), les échéances ainsi que les priorités.

Durant cette phase un autre document est produit : l'ICD – Interface Control Document qui répertorie l'ensemble des valeurs pouvant sortir du logiciel et y entrer (TC, TM, commandes OBDH, contenu de la mémoire de sauvegarde SGM) et spécifie leur format, taille, position des champs de données dans une trame, etc. Une version sous forme de tableau Excel est également produite durant la phase AD, c'est l'IDS, et ce document liste en particulier tous les mnémoniques qui seront utilisés lors du codage et de la validation.

La phase de spécification détaillée comprend également la création des packages Ada, fonctions et procédures (interfaces uniquement, les corps seront codés ultérieurement) incluant les commentaires décrivant chaque procedure ainsi que les références des exigences fixées dans la SRD que la procedure implémente. Cette tâche est effectuée en suivant la méthode HOOD – Hierarchical Object Oriented Design, développée en 1987 sous contrat de l'ESA. HOOD est une méthode de design de logiciels par voie graphique, associant diagrammes et texte. Le logiciel utilisé par Alcatel pour implémenter la méthode HOOD s'appelle STOOD. Il a été choisi en raison de son efficacité, sa robustesse et parce que c'est un produit "open source". STOOD permet de générer automatiquement les spécifications des packages Ada ainsi que le squelette des objets.

La production de cette phase est validée à l'issue d'une revue : l'ADR – Architectural Design Review.

5.1.3 Codage

La phase de codage, tout au bas du cycle en V, est la véritable réalisation du logiciel de vol. Cela consiste à « remplir » les squelettes créés durant la phase précédente, en se servant de tous les documents produits dans les phases de spécification. Cependant, le plus souvent il s'agit de modifier du code déjà existant, étant donné le caractère incrémental du développement logiciel, d'une part, et d'autre part la nécessité de modification fréquente du code, lorsque des erreurs sont décelées.

A l'issue de cette phase, un binaire du logiciel de vol est généré. Il servira notamment à la validation.

5.1.3.1 Ada 83

Le langage de programmation utilisé pour le codage du logiciel de bord est Ada 83.
Le langage Ada, destiné à la programmation de systèmes embarqués, a été créé dans le but de fournir non seulement un langage de programmation plus robuste, plus exact et plus sûr, mais aussi de rendre le

© Alcatel Alenia Space 2005

processus de développement et de maintenance (aussi dénommé génie logiciel) de logiciels critiques ou de grande complexité plus efficace, plus rapide et moins coûteux.

Il va sans dire que la programmation de satellites, correspond parfaitement exigences pour lesquelles le langage Ada a été élaboré ; les logiciels de bord d'engins spatiaux sont avant tout destinés à être embarqués, ils sont par ailleurs très critiques étant donné l'absence de possibilité de réparation directe d'un satellite. Et cela sans parler du milieu extrêmement hostile qu'est l'environnement spatial. Par ailleurs, les contraintes de durée de vie opérationnelle pouvant aisément dépasser la dizaine d'année, le logiciel se doit d'être parfaitement fiable. Un dernier point pour lequel Ada s'accorde bien aux systèmes satellites, est la relativement grande taille des projets et le nombre important d'ingénieurs se partageant le travail de développement et de mise à jour.

Les points fondamentaux permettant au langage Ada s'acquitter de ce cahier des charges sont :

- *La lisibilité du code source, ce dernier devant être lu (et relu) souvent et par beaucoup de gens, rendant ainsi la compréhension (par une personne autre que l'auteur du code source) et l'analyse respectivement plus facile et efficace.*

- *Le typage fort, permettant la manifestation de la majorité des erreurs et des confusions lors de la compilation plutôt que lors de l'exécution, (ou même pas du tout), comme c'est le cas dans de nombreux langages de programmation non fortement typés.*

- *La programmation à un niveau macroscopique, grâce aux mécanismes de compilation séparée, d'encapsulation, de masquage d'information et de gestion de bibliothèques.*

5.1.3.2 Clearcase

Le logiciel de vol étant d'une taille et complexité relativement importante, développé en parallèle par différentes personnes et de surcroît de façon incrémentale (chaque version se basant essentiellement sur la précédente), il s'avère rapidement indispensable de disposer d'un mécanisme de gestion des nombreux fichiers, chacun existant dans différentes versions. Dans notre cas, c'est le logiciel Clearcase qui s'acquitte de cette tâche. Il permet en particulier :

- *Le verrouillage d'un fichier par un utilisateur afin d'éviter toute modification concurrente, induisant des inconsistances. (mécanisme de check-in et check-out).*

- *L'attribution de numéros de versions aux fichiers modifiés. (Numéros de version propres à un fichier en particulier et non à l'ensemble de l'application). Les versions antérieures sont ainsi conservées en permanence, permettant d'y retourner au besoin, ou de les utiliser afin de mettre en évidence l'évolution.*

- *L'attribution de numéros (labels) propres à une version de l'application afin de garantir une cohérence entre les fichiers et permettre ainsi de réutiliser, recompiler et exécuter le logiciel tel qu'il était à un certain moment de son développement.*

- *Le masquage des fichiers non pertinents (au moyen du concept de vue). On crée ainsi un environnement de travail dans lequel seuls les fichiers de la version choisie sont visibles, évitant de cette façon le besoin de renommage « manuel » des fichiers. (les fichiers sont renommés au travers de Clearcase, par l'ajout d'une extension comprenant le numéro de version).*

- *La possibilité de représentation graphique des versions sous forme d'arbres, ce qui procure une meilleure vue de l'état de développement.*

© Alcatel Alenia Space 2005

- *La possibilité de comparer visuellement, en parallèle, deux versions d'un même fichier afin de rapidement se rendre compte des modifications qu'il a subi entre deux versions.*

- *La gestion de la production, en contrôlant automatiquement les dépendances.*

5.1.3.3 Règles de codage

Afin de conserver un standard de lisibilité, de qualité, et pour eviter des erreurs « classiques », et pour des questions fonctionnelles (prévision d'utilisation de mémoire, etc), des règles de codages, réunies dans un document dédié, sont à respecter lors du codage des procédures.

5.1.4 Validation

Examinons à présent la partie remontante du cycle en V. Celle-ci est consacrée à la validation du logiciel. Nous détaillerons uniquement la première phase, les tests unitaires, le stage n'étant pas concerné par les autres phases.

5.1.4.1 UT – Tests unitaires

C'est dans la réalisation de cette phase que la quasi totalité du stage s'est déroulé.
Les tests unitaires visent à s'assurer du fonctionnement correct de chaque package Ada constituant le logiciel de vol. On teste une fonction ou procédure à la fois, afin de s'assurer du bon fonctionnement de chacun des éléments constituant le logiciel. Le fonctionnement est déclaré correct lorsqu'une procédure invoquée avec un état initial donné (paramètres d'appels, variables globales, etc) produit l'état final escompté (paramètres de retour, nombre de procédures appelées, etc).

Il existe des logiciels spécifiques servant à effectuer des tests unitaires. L'un d'eux, ATTOL, est celui qui est utilisé par Alcatel Space. Il fournit un environnement facilitant la rédaction de tests et permet deux approches aux tests unitaires :

- *<u>Les tests d'intégration</u>, dont le principe est d'exécuter non seulement le code le la procédure à tester, mais également le code des fonctions et procédures appelées par la cette dernière. Cette approche a pour avantage de rendre la rédaction des tests plus simples, les procédures testées étant vues comme des « boites noires » dont toutes les actions « internes » sont ignorées par le testeur. Seules les valeurs de retour de la procédure testée sont examinées (dans certains cas il sera tout de même nécessaire d'observer les variables internes). En revanche, l'inconvénient est de s'exposer au risque d'erreur causé par les procédures en arrière plan, rendant ainsi la recherche des erreurs plus difficile.*

- *<u>Les tests utilisant des STUBs.</u> La procédure à tester peut être isolée du reste du logiciel dans le but de déceler uniquement les erreurs contenues dans cette partie de code. Ceci est rendu possible grâce au mécanisme de STUB, qui permet de supprimer l'execution du corps des procédures appelées et de paramétrer « à la main » les valeurs de retour. Toutes les interfaces de la procédure testée peuvent ainsi être observées et maîtrisées. Ces interfaces sont : le nombre d'appels à une certaine procédure, les valeurs des paramètres à l'invocation et les paramètres retournés. L'avantage de cette méthode, est de travailler sur des blocs de code plus courts, ce qui facilite la localisation des erreurs, s'il s'en produit. De plus, les erreurs au niveau des appels de procédures sont aisément décelables, les interfaces étant toutes surveillées. L'inconvénient est le fait qu'il faille spécifier « à la main » tous les paramètres d'interfaces lors de la rédaction du test. Ce travail peut devenir très fastidieux lorsqu'il s'agit de calculs compliqués ou d'appels de procédures nombreux (par exemple, appels de procédures dans une boucle effectuant un grand nombre d'itérations).*

Le choix de l'une ou l'autre des approches incombe au rédacteur du programme de test et est une question de jugement, mettant en balance, entre autres, la fiabilité (estimée) des procédures appelées, la taille (en nombre de lignes de code) de ces procédures, la complexité de la rédaction manuelle des paramètres d'interface.

© Alcatel Alenia Space 2005

La philosophie du test unitaire est de procéder de façon fonctionnelle, en utilisant les procédures à tester telles des boites noires, sans se préoccuper du comportement interne, en vérifiant uniquement que les fonctions (ou actions) sont correctement exécutées. Il s'avère parfois plus judicieux de tester conjointement plusieurs procédures, dont le test réellement unitaire imposerait l'examen de variables internes. L'exemple type est celui consistant à tester conjointement la procédure générique SET_VARIABLE_X (qui affecte une valeur à la variable X) et la fonction GET_VARIABLE_X (qui retourne la valeur de la variable X), en vérifiant que la variable donnée en paramètre d'entrée de la procédure SET_VARIABLE_X est bien celle qui est retournée par la fonction GET_VARIABLE_X.

Voici le déroulement d'une phase de tests unitaires ainsi que faite durant le stage :

1) *Compilation du code source Ada qui fera l'objet du test unitaire.*

2) *Rédaction du fichier de test ATTOL. Ce fichier, portant l'extension .ptu est codé dans un langage spécifique à ATTOL. Chaque fichier .ptu teste l'un des packages Ada du logiciel de vol. Un fichier de test comporte en principe plusieurs blocs SERVICE, un pour chaque procédure ou ensemble de procédures testée, chacun composé de un ou plusieurs blocs TEST. Les blocs TEST servent à spécifier un scénario d'exécution d'une partie du logiciel. On retrouvera ainsi généralement un ou pluieurs scénarios de fonctionnement « nominal » et différents scénarios présentant des cas d'erreurs. Le but étant de tester tous les cas possibles et d'executer toutes les parties du code. Un bloc TEST commence avec l'intialisation des variables dont on souhaite observer le changement (ou non) de la valeur résultant de l'exécution du scénario de test. Le bloc TEST se termine par l'invocation des procédures et fonctions à tester, directement en langage Ada, ce qui spécifie le scénario joué.*

   Lorsque des STUBs sont utilisés, il faut également spécifier les paramètres d'appels et les paramètres de retour. La spécification des paramètres d'appel permet de comparer les valeurs attendues aux valeurs effectivement passées en paramètre par la procédure testée. Les paramètres de retour sont choisis par le rédacteur du test, afin de spécifier le scénario de test. Un exemple de fichier de test .ptu est fourni en exemple dans l'annexe, accompagné de légendes.

3) *Preprocessing du fichier de test. Cette étape sert à générer un code source Ada implémentant le test unitaire. Une analyse lexicale et syntaxique est également réalisée et produit des messages d'erreur le cas échéant, nécessitant une correction du .ptu.*

4) *Compilation du fichier Ada.*

5) *Edition des liens.*

6) *Exécution du programme de test.*

7) *Création (automatique) du rapport de test. ATTOL génère un rapport rendant compte du résultat du test. Si les valeurs attendues ne correspondent pas aux valeurs obtenues, le test correspondant sera déclaré « incorrect ». Ce fichier porte l'extension .ro, un exemple est fourni en annexe.*

Les tests unitaires permettent également de tester la non-régression (conservation du comportement souhaité) de composants logiciels lorsque des modifications sont effectuées dans d'autres parties du logiciel.

Clearcase est également utilisé pour gérer les différentes versions des fichiers de test unitaires ATTOL (.ptu). Les labels (numéros de versions) correspondent avec les labels des fichiers source Ada, de telle façon qu'il est possible de rejouer les tests sur d'anciennes versions.

© Alcatel Alenia Space 2005

5.1.4.2 VT – Tests de validation

Le but de cette phase est de tester l'ensemble des fonctionnalités du logiciel de vol et de vérifier leur conformité aux exigences fixées dans la SRD.

L'équipe de validation commence par établir un plan de tests, contenant toutes les spécifications définies à partir de la SRD. Puis les procédures de tests sont codées dans un langage spécifique à cette tâche, le langage ATL qui est propre à Alcatel et est basé sur du C. Ce langage est indépendant des équipements de tests visés. Quelque soit le banc de test utilisé, le test ainsi codé pourra être joué.

La procédure de test définira d'une part les actions (temps réel) à effectuer (étape exécution), et d'autre part les contrôles des données désirées (étape analyse). Cette rédaction effectuée, les tests sont compilés et exécutés sur le banc de validation logiciel. Une fois l'execution des tests terminée, l'analyse des résultats est abordée.

5.1.4.3 DE – Demande d'Essai

Cette phase correspond aux tests effectués sur le programme lorsqu'il est implanté sur une cible avec les vrais équipements. En fait, chaque équipement va être testé avec le logiciel. Dans la phase précédente, le logiciel était testé avec des simulateurs d'équipements uniquement.

Il existe encore un dernier niveau de validation, n'ayant pas spécialement de correspondance avec une phase de conception : les Tests d'Intégrations (AIT). Cette phase ultime permet de valider tous les modules du satellite en salle blanche. C'est en quelque sorte, le test grandeur nature du satellite, entièrement assemblé, tel qu'il sera dans l'espace. Il est testé en condition.

Enfin, il est bon de noter que tous les documents produits (aux cours de ces différentes phases) subissent des modifications et des numérotations sous forme de numéro de version et de révision. Ceci permet de tracer le cheminement de la correction des erreurs et de progresser par paliers. A chaque modification, les anciens tests sont rejoués pour s'assurer que ces dernières ne modifient pas le fonctionnement validé précédemment.

5.2 Organisation

Le cycle en V nous permet de comprendre comment est produit le logiciel de vol. Ce n'est cependant pas suffisant pour comprendre tout le processus de développement logiciel, étant donné que le cycle n'est jamais exécuté d'un bout à l'autre, d'un seul tenant. Des rebouclages (retours en arrière lorsque des erreurs ont été décelées) sont en général nécessaires. Par ailleurs, les différents projets et satellites, reposant sur des bases identiques n'ont pas forcément des cycles de développement distincts les uns des autres, mais liés.

Tout d'abord il existe plusieurs projets en cours au même molent dans le service logiciel de vol. Un responsable de projet supervise l'ensemble des équipes. Il s'assure que les délais pourront être respectés, et organise le travail. Voici les différentes équipes ainsi que leurs tâches :

- *L'équipe système est responsable des études puis de la rédaction des spécifications, donc de toute la partie « descendante » du cycle en V.*

- *L'équipe développement se charge de l'étape « Codage » du cycle en V. C'est cette équipe que j'ai rejoint durant mon stage. Elle code les spécifications en Ada, effectue les tests unitaires et produit un binaire (une version du logiciel). L'équipe développement effectue par ailleurs de la relecture de code afin d'assurer une certaine qualité. Les remarques émises durant les relectures sont transmises à l'équipe qualité, qui décidera alors s'il convient d'effectuer les modifications sugérées.*

© Alcatel Alenia Space 2005

- *L'équipe validation s'occupe alors de tester ce binaire, conformément au plan de test qu'elle s'est fixée. En effet, au préalable, un plan de validation a été réalisé. Elle édite ensuite les rapports de tests qu'elle communique au responsable de projet et à l'équipe qualité. Si toutefois il existe des problèmes lors des tests (non respect des exigences, comportement inattendu du logiciel....), ils sont notifiés à l'équipe développement via des document appelés SPR – Software Problem Report. Les problèmes plus importants remonteront au niveau des études. Des document appelés SCR – Software Change Request – sont également émis lorsque des modification dans le code sont requises.*

- *L'équipe qualité, plus restreinte, a pour rôle de contrôler le respect des règles de qualité : version et numérotation des binaires correctes, référencement de documentation cohérent. Elle prend connaissance de tous les rapports de tests pour s'assurer qu'ils sont conformes aux exigences et aux divers documents de support.*

Au sein d'un même projet, la communication est importante, les équipes dépendant fortement les unes des autres. Des réunions sont régulièrement organisées, au moins une par semaine, afin de faire le point sur l'avancement du projet, de fixer les nouveaux objectifs, et pour mettre en relief et traiter les problèmes rencontrés.

© Alcatel Alenia Space 2005

6 Le stage

6.1 Intitulé du stage

Voici l'intitulé du stage réalisé, ainsi qu'il était présenté dans l'offre de stage :

PRESENTATION DE L'ACTIVITE DU SERVICE DEMANDEUR
Le stagiaire travaillera au sein du service BO/AV/LVL en charge de développer les logiciels vols des calculateurs bords des satellites de communication et d'observation. Le service développe des produits logiciel à forte contrainte temps réel, critique (niveau de fiabilité important), et à forte composante algorithmique pour certaines fonctions .

INTITULE ET DESCRIPTION DU SUJET DE STAGE ET DES RESPONSABILITES DU STAGIAIRE
Le stagiaire sera en charge à partir d'une spécification logicielle de concevoir une fonction majeure à embarquer dans le produit logiciel vol. Une fois cette conception agréée par le responsable développement, il sera en charge de coder et tester unitairement sa fonction. Ce stage lui permettra de prendre conscience des réalités du monde industriel en terme de développement d'un produit logiciel critique.

6.2 Objectif

L'objectif du stage était d'intégrer l'équipe de développement logiciel travaillant sur l'application de gestion de la batterie du satellite Koreasat V, satellite de la serie Spacebus 4000.

Le développement en cours à ce moment là était l'adaptation du logiciel de vol au remplacement d'un équipement de l'avionique : une carte électronique de gestion de la batterie (la carte BEU – Battery Electronics Unit) a été remplacée par une autre carte, la LMU – Lithium ion Management Unit. Ce changement impliquait un bon nombre de modifications, notamment au niveau du code Ada et des procédures de tests unitaires correspondants.

6.3 Travail réalisé

Le travail réalisé a consisté à la mise à jour des procédures de tests unitaires, les fichiers ATTOL (.ptu), afin qu'ils s'adaptent aux changements effectués dans le code source du logiciel de vol.

Pour donner une idée de la quantité de travail réalisé, il faut d'abord observer la taille l'application de gestion de la batterie (BMLI). Celle-ci est composée de 79 fichiers Ada (fichiers de spécifications et fichiers de corps de packages), représentant plus de 40'000 lignes de code (commentaires compris). En ce qui concerne les fichiers de tests unitaires, leur nombre est de 32, et la taille[1] approche les 95'000 lignes. Sur ces 32 fichiers, j'ai été amené à travailler sur 17 d'entre eux, ce qui implique une relecture (et un travail de compréhension) du code Ada testé. Parmi les 17, j'ai entièrement conçu deux des fichiers de test car pour deux packages Ada du logiciel de vol, le programme de tests unitaires n'existait pas encore. Ma tâche a donc été de concevoir les scénarios de tests et de rédiger à partir d'une feuille blanche le fichier de test

[1] *Au moment de la rédaction de ce rapport.*

© Alcatel Alenia Space 2005

ATTOL correspondant. Pour d'autres packages, les changements avec la version précédente du code source étant si importants, qu'il était plus simple de repartir depuis zero (en s'inspirant toutefois du scénario de test) plutôt que d'adapter la procédure de test existante.

Dans le cas d'une mise à jour du fichier de test, les modifications étaient de différents types :

- *Modification du type d'une variable.*
- *Modification de la valeur d'une constante.*
- *Changement du nombre d'appels à une procédure en STUB.*
- *Ajout d'appels de procédures.*
- *Modification due à un changement dans un algorithme.*
- *Changement de nom d'une variable, d'une procédure ou d'un type.*
- *Etc.*

Dans le cas d'une conception depuis zero d'un fichier de test, il s'agissait d'élaborer les tests de façon à verifier que le comportement effectif des procédures correspondait bien aux exigences, ainsi qu'expliqué dans la section relative aux tests unitaires. L'un des points clés était l'identification des différents cas d'erreur pouvant se produire afin de tester le comportement correct de chaque procédure dans une situation particulière.

L'exécution des tests unitaires m'a permis d'identifier certaines erreurs dans les sources Ada : des appels répétés à certaines procédures dans un cas particulier ont mis en évidence des oublis de réinitialisation de variable.

Contrairement à la description dans l'intitulé du stage, je n'ai pas eu à concevoir de fonction majeure à embarquer dans le produit logiciel de vol... Seules quelques évolutions de besoin ont été codées par moi-même, sur la base des modifications apportées à la SRD. Ces évolutions se traduisaient par la modification de quelques lignes de code seulement.

J'ai également participé à plusieurs phases de relecture de code, dont la première s'est déroulée peu après le début du stage. J'ai d'une part eu à vérifier que les règles de codage étaient respectées. Ces règles visant à assurer un standard de codage et à éviter des erreurs "prévisibles", elles permettent d'améliorer sa façon de programmer, notamment en acquérant des automatismes (servant par exemple à éviter des confusions entre variables globales ou locales, pointeurs ou éléments pointés). D'autre part, l'aspect fonctionnel doit bien sûr être vérifié. Ceci ammène à se poser les "bonnes questions". Il faut alors se référer à la documentation de spécification, ce qui permet ensuite de mieux saisir le fonctionnement du système (dans notre cas, le système de puissance électrique). La relecture de code m'a donc, en résumé, permis de "m'imprégner" du style de programmation en vigueur dans le service, avant de moi-même me lancer dans le codage.

© Alcatel Alenia Space 2005

7 Conclusion

En conclusion, l'analyse du travail effectué me permet de faire ressortir ce que le stage m'a apporté ainsi que les points qui ont représenté une difficulté. Les difficultés obligent à améliorer son niveau de compétence afin de les surmonter. Un stage sans aucune complication n'aurait qu'une très faible valeur éducative. La conclusion se termine par mes impression personnelles sous forme d'un bilan.

7.1 Difficultés rencontrées

Les principales difficultés rencontrées durant ce stage ont été :

- *La compréhension du produit à tester. La taille et le nombre de fichiers Ada constituant l'application de gestion de la batterie étant relativement importants, additionnés à toute la documentation, il m'a fallu un certain temps (environ un mois), avant de véritablement commencer à saisir le fonctionnement de tous les engrenages et avant de pouvoir devenir opérationnel. Pour fixer un ordre de grandeur, je rappelle la taille des documents concernés : du côté de la documentation, l'URD et la SRD comptent chacune une centaine de pages. Pour ce qui est du code, l'ensemble de fichiers Ada pour la gestion de la batterie totalisent plus de 40'000 lignes. Il n'est certes pas nécessaire d'en lire chaque partie, mais cela donne une idée de la complexité. Ensuite, les fichiers concernant les tests unitaires, représentent en moyenne le double de volume par rapport au code source testé. A tout cela est venu s'ajouter la lecture du manuel d'utilisateur du logiciel ATTOL. La difficulté d'entrer dans le projet est en fait directement liée au type de produit : il est critique (il ne doit subsiter aucune erreur lorsqu'il est délivré), industriel, et de grande taille.*

- *La documentation importante. Ainsi que décrit dans le point précédent, la taille des documents est considérable, lorsqu'on débute dans un projet industriel. Cette documentation, même si elle est initialement difficile à appréhender, constitue par la suite un véritable support sur lequel on s'appuye, notamment pour la compréhension du projet. La confrontation avec la documentation est assurément un point qui constitue une valeur ajoutée apportée par ce stage. En effet, durant mes études et mes stages précédents, je n'ai pratiquement pas eu à me livrer à cet exercice, la taille des projets, leur complexité et le nombre de personnes travaillant dessus étant à ce point inférieurs qu'ils ne nécessitaient pas véritablement de support documenté. Outre la taille de documentation, la quantité d'acronymes (certes explicités dans le glossaire) dont il est parfois difficile de se souvenir de la signification, a présenté une difficulté de plus. En revanche, il est bien clair que les acronymes sont introduits dans un but de clarté et de simplification. Cette difficulté s'estompe avec le temps, les termes utilisés fréquemment devenant familiers.*

- *Le typage fort introduit par Ada. Ce typage oblige parfois le programmeur à manipuler une dizaines de fichiers quasi simultanément, lorsqu'il s'agit d'aller retrouver par exemple le type d'une variable définie et redéfinie plusieurs fois (en utilisant le mécanisme de renames Ada). Ceci est inhérent à la politique de programmation et de gestion des interfaces choisies pour ce logiciel de vol. Petit à petit, on parvient néanmoins à développer des techniques pour s'acquitter de cette tâche avec plus de rapidité, d'habilité et d'efficacité. Ces techniques deviennent un acquis, réutilisable dans quelque situation que ce soit. Avec les autres langages que j'ai utilisé par le passé (Java, C, C++, etc), exempts de typage fort, cette caractéristique de références "traversant" les fichiers était absente.*

- *Les tests unitaires mal commentés. Il peut être très difficile de comprendre certains scénarios joués dans les tests unitaires, lorsqu'ils ont été rédigés par quelqu'un d'autre, si les commentaires explicatifs sont en quantité insuffisante par rapport au niveau de complexité du scénario. Il arrive même parfois de trouver des des commentaires contradictoires avec le code, résultat d'une mégarde ou d'une erreur. Ceci m'a laissé me rendre compte de l'importance des*

© Alcatel Alenia Space 2005

commentaires, des explications et de l'attention qu'il convenait de respecter lors de la rédaction. J'ai alors pris soin de décrire de façon claire mais concise les opérations effectuées ainsi que les intentions et les buts recherchés.

7.2 Compétences acquises

Et voici les points qui ont constitué une nouveauté pour moi :

- *Le langage Ada. Je n'avais jusqu'à présent jamais été confronté à ce langage. Ceci dit, étant habitué à la programmation, mis à part les points cités dans la section précédente, cela n'a pas représenté une réelle difficulté. Le caractère rigoureux du langage Ada (du fait de son élaboration afin d'être utilisé sur des systèmes critiques) est fort instructif car il met en garde et "dénonce" même d'autre langages (C, C++, Pascal, etc) en citant des exemples dans les livres de référence Ada, du danger ou risque d'erreurs possibles avec ces derniers langages, mais impossibles avec l'Ada. Tout cela renseigne sur les points à surveiller en programmation, dans quelque langage que ce soit et augmente la vigilence.*

- *L'architecture logicielle d'un satellite. Je possède à présent une meilleure idée de la façon dont est construit un logiciel embarqué sur un satellite. L'architecture système, ou fonctionnelle, a été largement abordée durant les cours de mastère, mais sans rentrer dans les "boites noires" que sont les équipements tels que l'ordinateur de bord. Cela aurait fait l'objet d'un module à part entière. Ce point constitue un complément très intéressant aux enseignements reçus durant la scolarité.*

- *La lecture d'une importante quantité de documentation. Ainsi que développé dans la section précédente, ce point qui s'est initialement présenté comme une difficulté, s'avère être une des nouveautés les plus importantes que m'aura apporté ce stage.*

- *L'utilisation de logiciels de gestion de versions et de configurations. J'ai appris les rudiments d'utilisation de Clearcase. Ce type de logiciel, est, je l'imagine, utilisé dans tous les types de développement informatique industriel. Les notions que j'ai acquises me seront sans aucune doute très utiles dans le futur.*

- *L'utilisation de logiciel dédié aux tests unitaires (ATTOL). J'avais déjà, durant un autre stage, réalisé des tests unitaires, mais en concevant intégralement le programme de test, sans utiliser d'infrastructure préexistante. En lisant la documentation ATTOL et en manipulant le logiciel, j'ai été sensibilisé à un certain nombre de particularités propres aux tests unitaires que je n'avais pas rencontré par le passé et dont je n'étais pas conscient.*

7.3 Bilan

Ce stage m'a permis de participer activement au développement d'un engin spatial et de ce fait, d'atteindre le but que je m'étais fixé. Il m'a par ailleurs fourni un aperçu des réalités du développement industriel, de ses outils informatiques, de ses processus, du travail en équipe, de la documentation indissociable du projet, consolidant ainsi mon expérience professionnelle, d'une grande importance en vue de l'obtention de mon premier emploi.

Cependant, j'aurais souhaité pouvoir mettre plus à profit les connaissances en ingénierie astronautique acquises durant les six mois de cours de mastère à Toulouse. Ce stage était en effet bien plus orienté vers l'informatique que vers les techniques spécifiquement liées aux activités spatiales, telles que l'analyse de mission, la conception "système", etc. Je regrette également quelque peu le manque de diversité

© Alcatel Alenia Space 2005

et de possibilités d'initiatives dans la nature des activités qui ont constitué mon stage, ceci étant certainement du au caractère critique du produit que j'avais à tester.

Je retiendrai néanmoins de ce stage une expérience très positive sur laquelle je pourrai m'appuyer pour la suite de ma carrière et de mes choix professionnels.

© Alcatel Alenia Space 2005