



BTS Informatique de gestion : Développeur d'applications



Note de synthèse

Projet Corrinos

Table des matières

TABLE DES MATIERES.....	2
REMERCIEMENTS	3
PRESENTATION DE LA SOCIETE	3
<i>Présentation de la société cliente.....</i>	<i>3</i>
MON ACTIVITE DANS L'ENTREPRISE	4
LE PROJET CORRINOS	5
<i>Problématique</i>	<i>5</i>
<i>Etude de l'existant.....</i>	<i>7</i>
<i>Les besoins.....</i>	<i>8</i>
Priorité des machines.....	8
Fonctionnalités de Corrinos	8
<i>Les contraintes</i>	<i>8</i>
<i>Déroulement du projet</i>	<i>8</i>
Infrastructure	8
Développement.....	10
Conception de la base donnée	10
Conception de l'interface	11
Définition de la couche d'accès aux données.....	13
Ajout des options de configurations.....	13
Intégration Corrinos dans Arrakys.....	14
Ajout du panneau de recherche	14
Problèmes rencontrés	15
Le transfert des programmes directement vers les machines	15
Restriction de l'accès au répertoire de centralisation	15
Sécurité relative à la conservation de l'identifiant	16
<i>Test</i>	<i>16</i>
Fiabilité du réseau wifi de l'atelier	16
Utilisation de l'application	17
<i>Le déploiement</i>	<i>17</i>
<i>La formation</i>	<i>17</i>
<i>Améliorations possibles.....</i>	<i>18</i>
Augmenter le nombre machines-outils compatibles avec le transfert de Corrinos	18
Gestion de la zapette	18
Intégration d'un éditeur de programme	18
CONCLUSION	19

ANNEXE1 Propositions d'infrastructures

ANNEXE 2 Documentation Utilisateur

Remerciements

Je tiens à remercier tout particulièrement l'intégralité de l'équipe de Dune pour avoir facilité mon intégration dans la société et pour avoir su être accessible tout au long de ma formation en BTS, dont mon maître d'apprentissage M. Nicolas Roche pour tout le savoir offert.

Je tiens également à remercier :

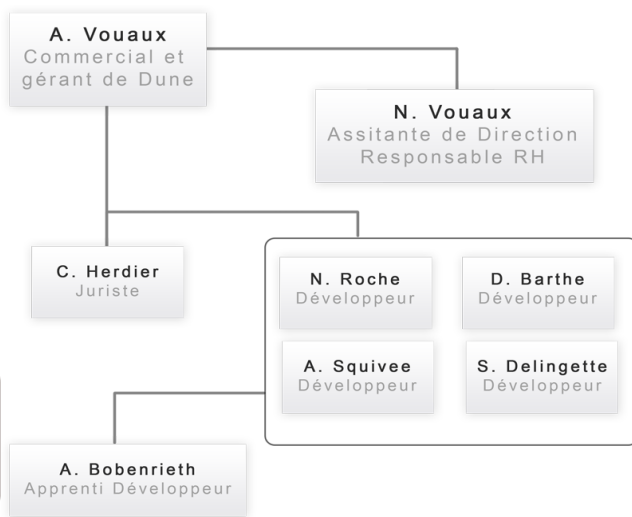
- notre client RML pour sa confiance
- mon intermédiaire dans ce projet M. Philippe Cousin pour sa disponibilité et sa clarté
- l'ensemble du CFA Robert Schuman pour la qualité du cadre fourni et de l'enseignement dispensé

Présentation de la société

Dune est une société lorraine créée en mai 2009, spécialisée dans le développement informatique, la formation industrielle et le suivi de projets innovants. Elle se distingue par une étroite collaboration avec les PMI, propice à la conception d'applications toujours plus spécifiques aux corps de métier des clients.

Ses produits principaux sont les solutions Arrakys et Caladan, qui fournissent un système de gestion de production et de relation fournisseur complet constamment en évolution pour suivre les nouveaux besoins de chaque client.

Elle emploie 9 personnes dont 5 développeurs, qui se concentrent dans l'exploitation des technologies Microsoft (vb6, .NET) pour la réalisation des produits. Outre le suivi des projets innovants, le service juridique composé d'un juriste a pour but de trouver de potentiels fonds d'aides dans la réalisation des nouveaux projets.



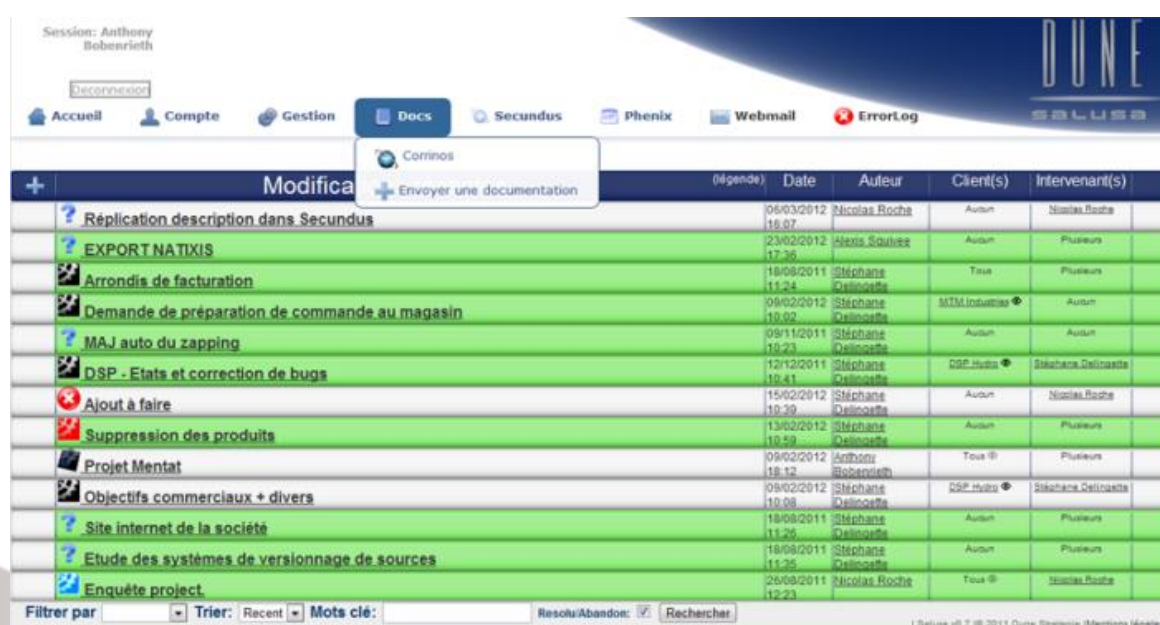
Présentation de la société cliente

La société RML Micron-Est est une entreprise lorraine située à Florange et fondée en 1986. Elle se spécialise dans la réalisation de pièces de mécaniques de précision. Elle se démarque par sa grande rigueur de gestion et de respect des normes industrielles ainsi que par la complète informatisation du processus de production.

Mon activité dans l'entreprise

Voici une brève présentation de mes principaux projets réalisés au cours de ces deux ans (énumérés de façons chronologique).

J'ai débuté mon apprentissage chez Dune par la réalisation d'outils internes avec, dans un premier temps, un gestionnaire de demandes d'améliorations (réalisé en PHP/HTML/JS*) pour les produits Dune, qui sera nommé par la suite Salusa. Orienté autant client que développeur (avec des règles de gestion de confidentialité des publications) elle permet la remontée manuelle de bug mais également des besoins de nouvelles fonctionnalités.



The screenshot shows the Salusa application interface. At the top, there's a session bar for 'Anthony Bobenrieth' with a 'Déconnexion' button. Below it is a navigation menu with 'Accueil', 'Compte', 'Gestion', 'Docs', 'Secundus', 'Phenix', 'Webmail', and 'Error.log'. A 'Corrinos' dropdown menu is open under 'Docs', showing 'Envoyer une documentation'. The main area is a table titled 'Modifica' with columns: (légende), Date, Auteur, Client(s), and Intervenant(s). The table lists various tasks with their dates, authors, and clients. At the bottom, there are filters for 'Filtrer par', 'Trier: Recent', 'Mots clé:', 'Résolu/Abandon: [X]', and a 'Rechercher' button. The footer indicates 'Salusa v0.7.00 2011 Dune Strategie (Attention légende)'.

(légende)	Date	Auteur	Client(s)	Intervenant(s)
?	06/03/2012 16:07	Nicolas Roche	Auton	Nicolas Roche
?	23/02/2012 17:36	Hens Soutiez	Auton	Plusieurs
?	18/08/2011 11:24	Stéphane Delcoiffe	Tous	Plusieurs
?	08/02/2012 10:02	Stéphane Delcoiffe	MTM Industries	Auton
?	09/11/2011 10:23	Stéphane Delcoiffe	Auton	Auton
?	12/12/2011 10:41	Stéphane Delcoiffe	GSP Hubs	Stéphane Delcoiffe
?	15/02/2012 10:39	Stéphane Delcoiffe	Auton	Nicolas Roche
?	13/02/2012 10:59	Stéphane Delcoiffe	Auton	Plusieurs
?	09/02/2012 18:12	Anthony Bobenrieth	Tous	Plusieurs
?	09/02/2012 10:08	Stéphane Delcoiffe	GSP Hubs	Stéphane Delcoiffe
?	18/08/2011 11:26	Stéphane Delcoiffe	Auton	Plusieurs
?	18/08/2011 11:26	Stéphane Delcoiffe	Auton	Plusieurs
?	26/08/2011 12:23	Nicolas Roche	Tous	Nicolas Roche

Cet outil deviendra par la suite l'extranet (site web sécurisé d'une entreprise, accessible depuis l'extérieur) de la société avec l'intégration d'une base de connaissance sous forme de wiki (documentation collaborative), un gestionnaire de remontée automatique d'erreurs (toujours réalisé en PHP) ainsi que l'intégration des fonctionnalités déjà existantes telles que l'agenda ou le webmail.

J'ai été chargé plus tard vis-à-vis de mes acquis en développement web de créer le site web de présentation de la société (en HTML/JS/PHP*) pour continuer à promouvoir les services de notre société. Essentiellement tourné sur le design, les seules parties de programmation pure concernant uniquement le formulaire de contact et la création du diaporama.



(Interface du site de présentation de la société)

J'ai également eu l'occasion d'avoir des interventions succinctes sur les autres projets en cours de la société. En outre, j'étais sollicité pour la « découverte » et la documentation de certains outils (comparaison de systèmes de versionnage, documentation WPF, mise en place d'un VPN...).

Tout au long de ces tâches, j'ai également réalisé en autonomie différents outils pour l'entreprise telle qu'une bibliothèque (recueil de « fonctionnalités » déjà codées) Dune, ainsi qu'un système de gestion de licences des produits Dune (Dune License Service).

Le projet présenté ici (Corrinos) est donc mon premier projet réalisé suite à la demande d'un client.

Aujourd'hui, je travaille surtout sur un futur système de GED (gestion électronique de document) qui se distinguera par sa faculté à lier les documents à des entités quelconques venant de diverses bases données, afin que cette GED soit adaptable à la plupart des logiciels professionnels, dont les petites applications qui ne sont pas supportées par les grandes GED d'aujourd'hui.

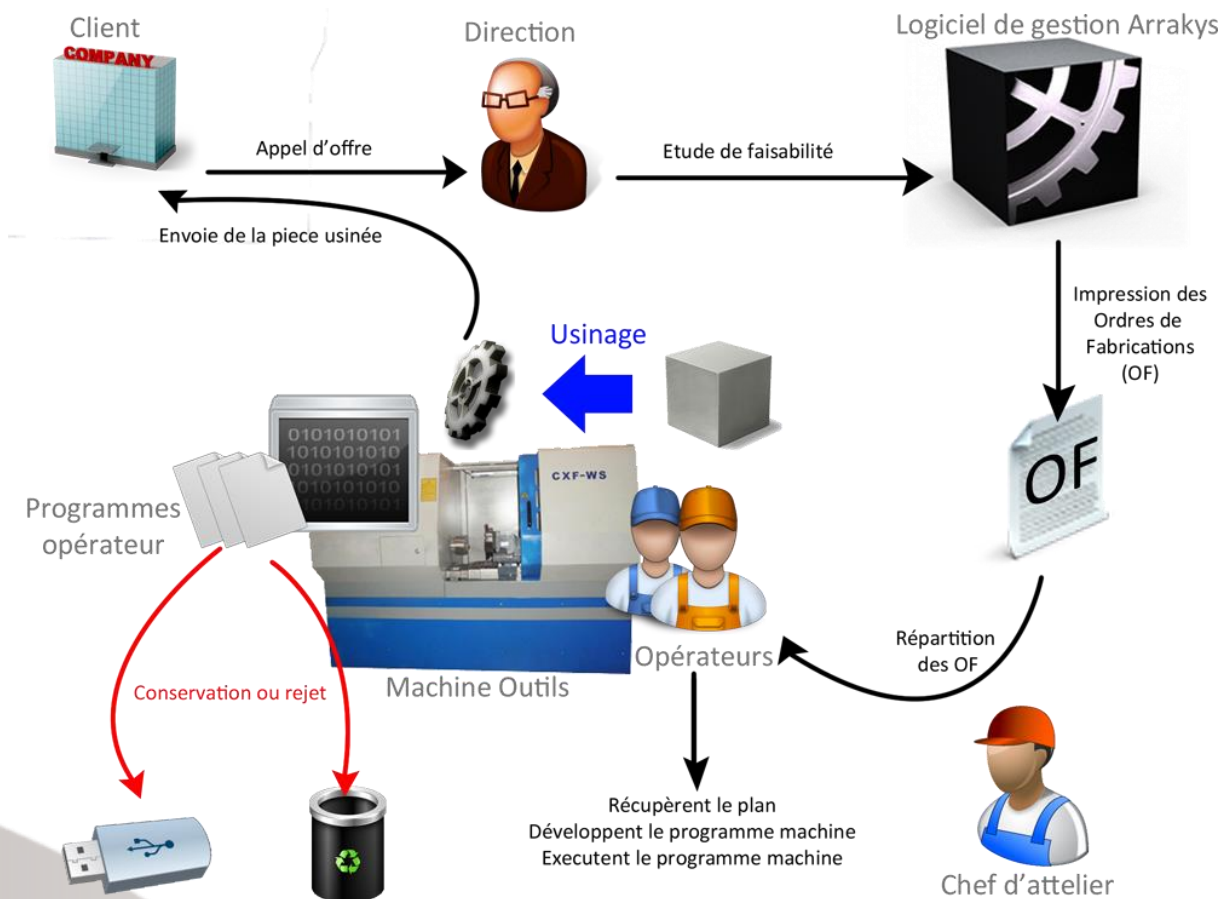
Le Projet Corrinos

Problématique



La réalisation de pièces de mécanique de précision se distingue par l'emploi de machines de tournage et fraisage semi-automatiques qui agissent selon les consignes données par les opérateurs de l'atelier. Ce sont les **machines-outils** à commandes numériques. On en compte une trentaine dans l'atelier de RML.

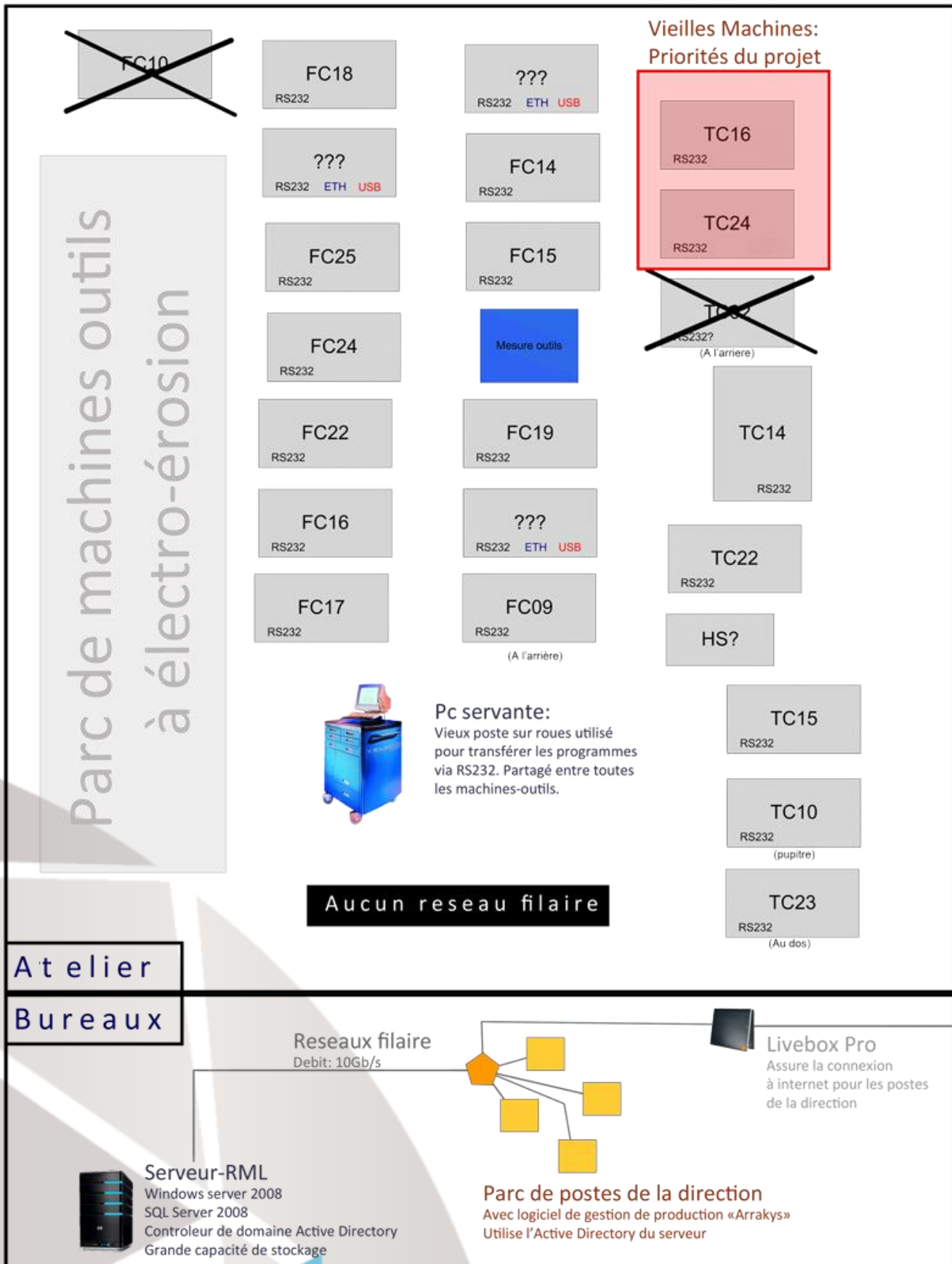
Loin des grandes chaines automatiques entièrement robotisées, ces machines réagissent à des saisies (qui constituent un **programme**) permettant à l'entreprise de s'adapter à des commandes très spécifiques pour des clients exerçant dans différents domaines tels que l'automobile ou l'aéronautique.



Toutefois, on relève de grandes similitudes entre les différents programmes ainsi qu'une récurrence d'utilisation de certains. L'espace de stockage limité qu'embarquent les machines-outils ne permet pas la conservation de tous les programmes, et il est souvent nécessaire aux tourneurs fraiseurs de parcourir l'ensemble du parc pour retrouver un vieux programme semblable à celui nécessaire pour une nouvelle réalisation, et parfois, de réécrire sa totalité une nouvelle fois.

L'idée a donc été initiée mi 2011 par L.Betti (Directeur General de RML) de réaliser un système de centralisation des programmes pour Machines-Outils, afin de réduire la perte de rendement de ses employés liée à la disparition des programmes.

Etude de l'existant



Les besoins

Priorité des machines

Le projet portera en priorité sur les machines de fraisage et de tournage, et plus particulièrement sur les vieilles machines de tournage disposant de moins de mémoire et d'aucune interface récente de stockage, obligeant les opérateurs à réécrire en continu leurs programmes.

Celles à Electro Erosion sont plus récentes et embarquent directement un système Windows (qu'on pourra intégrer plus facilement au projet Corrinos dans le futur). Elles ne sont pas directement concernées par le projet.

Fonctionnalités de Corrinos

Définies à l'issue d'une réunion, Corrinos devait en priorité permettre la centralisation des programmes sur le serveur de RML, en les liant à la référence plan de l'ordre de fabrication de l'opérateur et ainsi que permettre leur récupération à partir d'une référence plan donnée ou par l'intermédiaire d'une recherche basée sur plusieurs critères. On notera que son utilisation sera réservée aux opérateurs de l'atelier.

*Ordre de fabrication ou OF : Fiche relative à un traitement à effectuer par l'opérateur sur un élément d'une commande. Elles sont imprimées depuis le logiciel de gestion et sont réparties par le chef d'atelier aux tourneurs fraiseurs.

Côté administration, il est demandé que le chef d'atelier ainsi que la direction puisse voir au sein du logiciel de production de Dune « Arrakys » si telle ou telle pièce commandée a déjà un programme lié à sa référence.

Les contraintes

Aucune forte contrainte n'a été formulée par le client, toutefois les orientations du projet devaient lui être apportées et avoir sa validation. J'étais sur place une fois par semaine et disposais d'un plein accès à l'atelier (tout en prenant garde à ne pas déranger les employés dans leur travail).

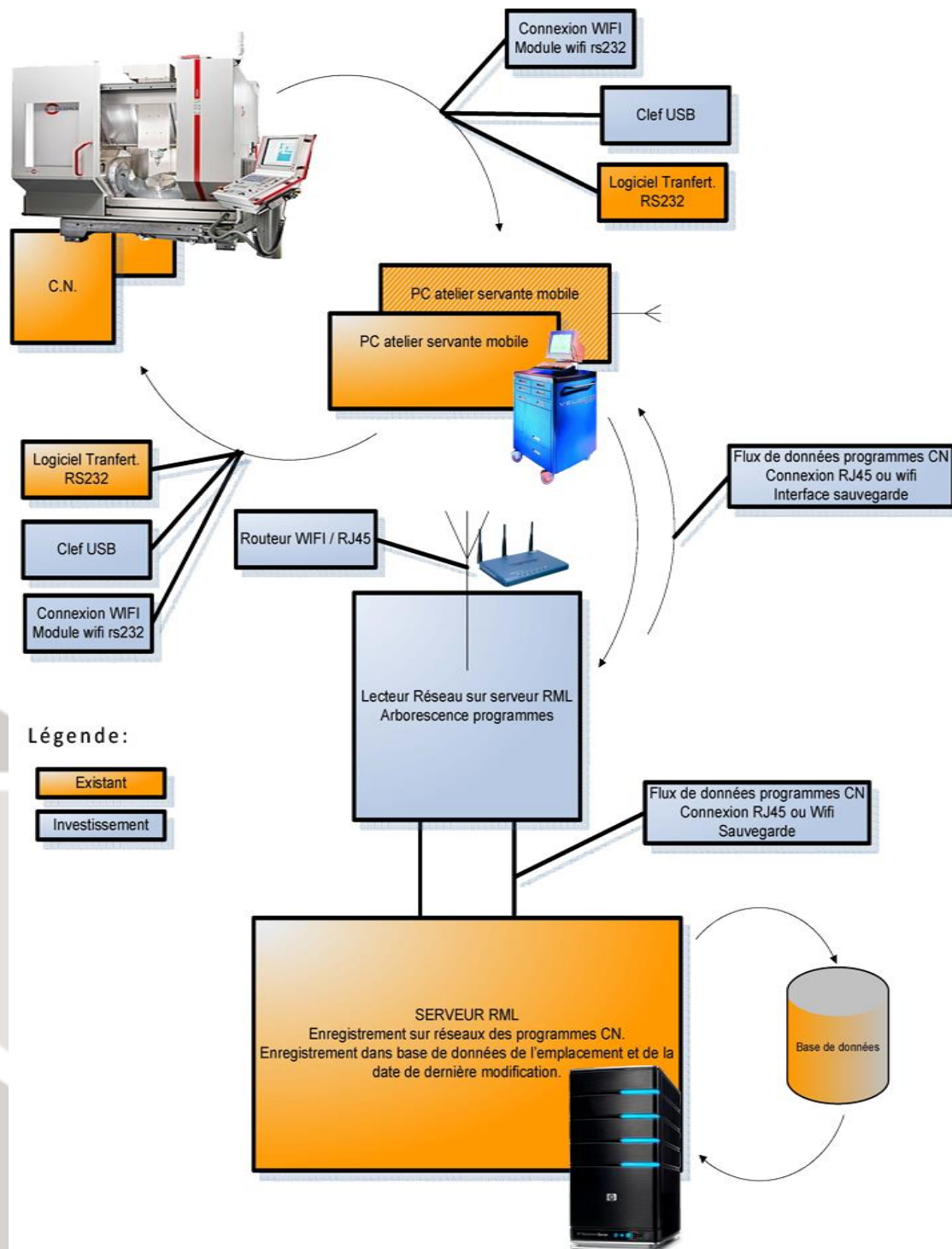
Déroulement du projet

Infrastructure

En première étape, il était important de réfléchir au matériel supplémentaire nécessaire à la mise en réseau des machines-outils et pour cela de faire une étude de l'existant. Côté bureaux, les infrastructures sont déjà connues de ma société, avec entre autres un réseau filaire disposant d'un débit de 10Gb/s ainsi qu'un serveur récemment installé qui servira au stockage des programmes ainsi que de support à la base de donnée.

Toutefois un bilan a été nécessaire côté atelier. Comme cela a été évoqué lors de la réunion, les interfaces varient d'une machine outils sur l'autre avec comme seul standard les ports série rs232, et aucune infrastructure réseau n'y est présente.

Nous étudions alors les possibilités d'infrastructure avec le responsable technique que nous représentons sous forme de schémas (fournis en annexes) avant de les transmettre à la direction pour délibération.



L'hypothèse d'acheter les extensions permettant l'ajout de ports aux machines est tout de suite écartée par le responsable technique de RML, car, beaucoup trop rares et coûteuses. On partira donc du principe que l'interface de base du projet se fera par le port rs232 seul interface concordant avec toutes les machines.

Hélas, l'utilisation de ce port interdit une intégration directe au réseau. En effet le port série est exclusivement réservé à l'utilisation de protocoles propriétaires nécessitant l'emploi d'une application spécifique au transfert de fichier. Un poste intermédiaire sera donc irrémédiablement nécessaire (nouvelles servantes ?) et ce, malgré les différents essais réalisés dans le but d'intégrer l'une des machines-outils au réseau à l'aide d'un convertisseur rs232/wifi...

Au vu des coûts requis à un câblage de l'atelier, la direction de RML préférera se tourner sur l'utilisation du wifi pour mettre en relation les nouvelles servantes et le réseau du serveur.

Cela impose donc l'achat d'un routeur wifi (type industriel, réputé pour être moins sensible aux interférences) ainsi qu'une nouvelle flotte de pc servantes équipées de port rs232 et d'interface wifi et qui seront portables cette fois (pour des questions de mobilité entre les machines). Le prestataire informatique de RML, la société MAI, se chargera de fournir l'ensemble de ces nouveaux équipements.

Développement

Compte tenu de la forte utilisation des outils Microsoft dans mon entreprise et de leur forte réputation dans l'industrie locale j'emploierai le langage C# pour la réalisation de Corrinos. *Visual studio* vient alors naturellement comme environnement de développement, et je profite de ce projet pour créer une instance de Team Foundation Server sur le serveur de Dune afin de versionner* mes sources.

Versionnage : Pratique de développement informatique permettant la centralisation et la sauvegarde des modifications faites au code source d'un projet. Très utile en collaboratif où la fusion des modifications effectuées par chaque développeur est simplifiée. De manière plus générale, la conservation d'un historique des modifications de ce genre d'outil permet de revenir à une précédente version du code afin de faciliter la localisation de potentiels nouveaux bugs ajoutés ou de toujours avoir accès à une version dite « stable » du code.

Conception de la base donnée

Elle sera directement intégrée à la base de donnée « Sorinter » du logiciel de gestion « Arrakys » développé par ma société.

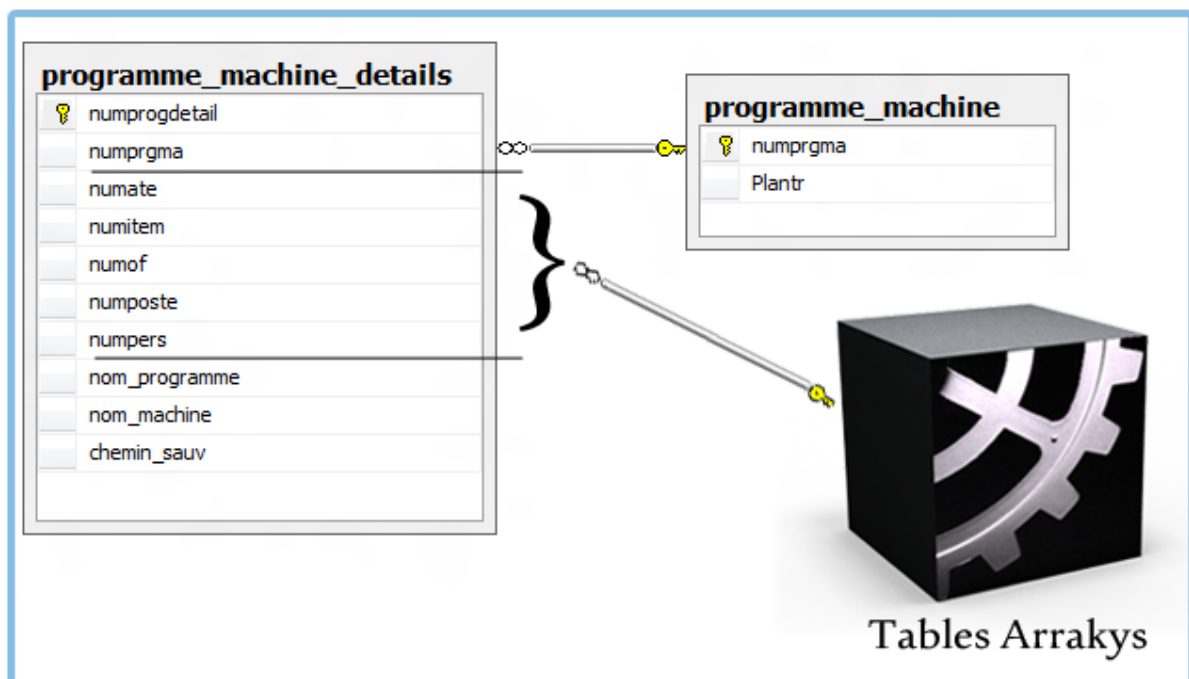
Plusieurs choix sont possibles, inscrire les programmes opérateurs directement dans la base de données (au format blob) ou les sauvegarder directement dans le système de fichier du serveur. Pour simplifier l'exploitation des programmes en cas de plantage de l'application, j'ai préféré utiliser directement le système de fichier de l'ordinateur et réserver l'utilisation de la base de

données uniquement au stockage des informations relatives des programmes opérateur sauvegardés.

Si dans un premier temps j'enregistrais les informations pour chaque fichier de programme copié (nom, taille), je me suis ensuite limité (sous conseil de mon tuteur) à référencer par lot de programmes reçus, en conservant le chemin de stockage du lot, l'application cliente se chargeant elle-même de faire une analyse du contenu du lot en explorant directement sur le serveur.

Cette exploration répétée n'est pas une menace de saturation du réseau du fait son fort débit (10Gb/s) et cela allège la quantité d'informations stockées dans la base de donnée.

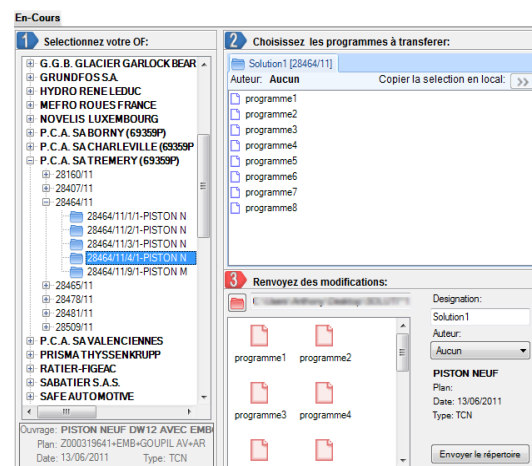
Le schéma conceptuel devient alors :

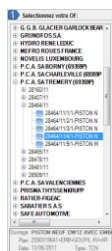


Base de données Sorinter

Conception de l'interface

Créer une application simple et intuitive aide grandement à sa démocratisation. Dans le cas de Corrinos j'ai décidé de décomposer son utilisation principale en trois étapes :





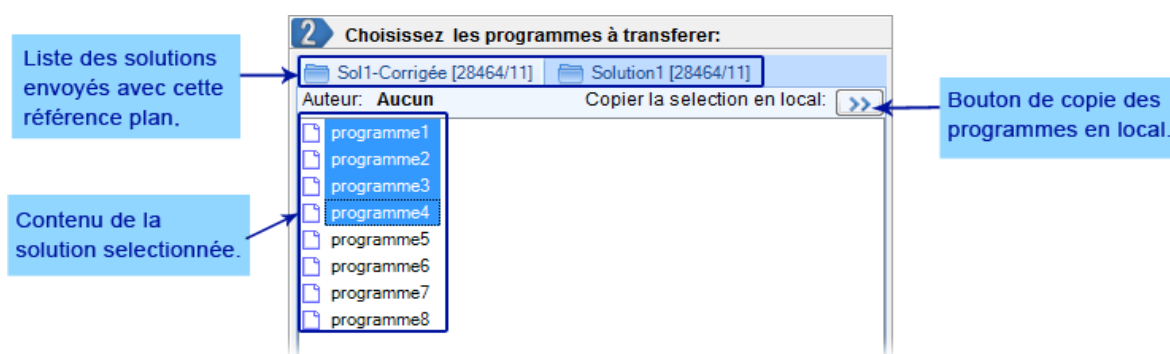
1 La sélection de l'ordre de fabrication (OF)

Elle est faite à partir du numéro inscrit sur l'ordre de fabrication papier de l'opérateur (les OF étant répartis en début de journée par le chef d'atelier).

C'est à partir de ce numéro d'OF que l'on détermine la référence de plan, et que l'on recherche la moindre concordance avec celle d'un programme opérateur existant.

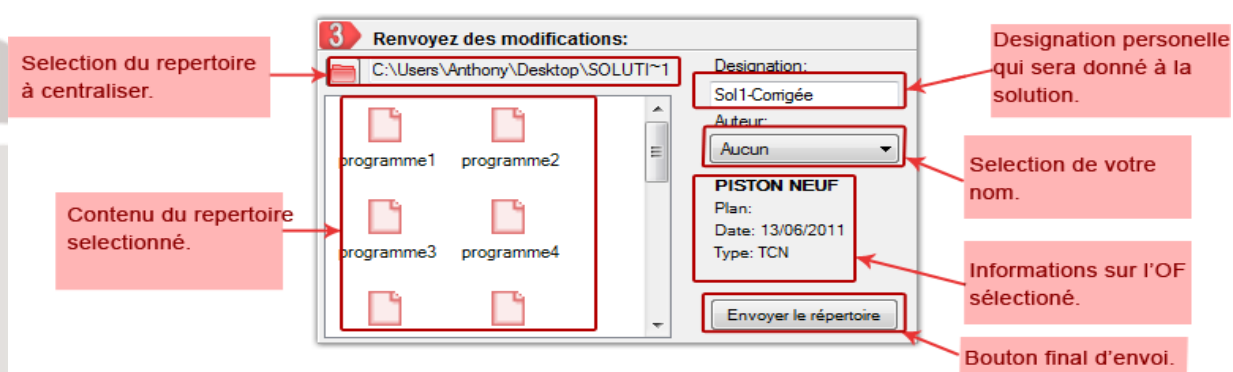
2 Récupération des programmes antécédents

Si des concordances sont trouvées, on liste les lots trouvés sous forme de répertoires affichés par onglet en listant les fichiers qu'ils contiennent. Tout ceci en explorant le système de fichier du serveur avec le client Corrinos de la servante.

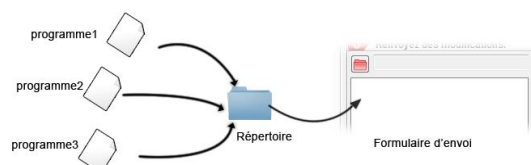


3 Renvoi des programmes modifiés

Après récupération et modification ou création de nouveaux programmes, l'opérateur est invité à renvoyer ses programmes sous forme de lot



Pour écourter et faciliter l'utilisation de Corrinos, on essaiera de multiplier les manières d'effectuer chacune de ces tâches pour multiplier nos chances qu'au moins une d'elles corresponde aux réflexes d'utilisation de l'opérateur courant. En intégrant par exemple une gestion de drag n drop* depuis l'environnement Windows des fichiers à renvoyer. Ou encore, en acceptant tout autant un répertoire qu'une sélection multiple de fichiers pour le renvoi... L'application devient alors plus intuitive et moins contraignante.



Définition de la couche d'accès aux données

La couche d'accès aux données est grossièrement la partie du code qui s'occupe de la transition d'informations d'une source extérieure (dans notre cas le server SQL) vers l'application. Elle dialogue généralement côté application avec des « classes métiers » censées représenter, sous forme d'objets manipulables par le développeur, les différents éléments de la base (des clients, des commandes...).

Débutant en C# je n'avais encore jamais eu à interagir réellement avec une base de données, or mes collègues avaient, à l'époque, commencé quelques projets de test utilisant l'Entity Framework, un outil de développement intégré à la plateforme .NET depuis 2008 permettant de générer cette couche d'accès aux données à partir d'un diagramme ou d'une structure de base de données existante.

Cette couche générée dispose de son lot de « classes métiers complètes » qui peuvent être utilisées directement dans le code sans se soucier des techniques de récupérations et de modifications. En effet l'accès et la modification de ces données est associé à un moteur qui s'occupe lui-même de traiter cet échange au moment nécessaire. Cela simplifie donc grandement l'accès aux données dans le code :

Méthode habituelle

```
BDD.Connexion();
string[] utilisateurEnPagaïlle =
    BDD.EnvoyerRequete("SELECT * FROM personnel");
List<Personnel> utilisateursOrdonnés =
    Personnel.OrdonnerResultat(utilisateurEnPagaïlle);
BDD.Deconnexion();
```

Méthode avec l'Entity Framework

```
List<Personnel> utilisateurs
    = BDD.personnel;
```

De plus j'ai couplé son utilisation à celle des requêtes LINQ qui permettent de filtrer directement dans le code la sélection des données qui seront reçues (ceci, en influençant les requêtes SQL générées par l'Entité Framework). Cette utilisation est en plus compatible avec l'IntelliSense de Visual studio, ainsi il est possible d'avoir une liste de l'ensemble des tables et champs et de bénéficier d'une analyse des requêtes révélant les erreurs avant l'exécution :

Méthode habituelle

```
BDD.Connexion();
string[] bosseursEnPagaïlle =
    BDD.EnvoyerRequete(
        "SELECT * FROM personnel pe "
        +"INNER JOIN programmes pr on pr.numbers = pe.numbers "
        +"WHERE COUNT(numprog)>1000");
List<Personnel> bosseursOrdonnés =
    Personnel.OrdonnerResultat(bosseursEnPagaïlle);
BDD.Deconnexion();
```

Méthode avec l'Entity Framework

```
List<Personnel> lesBosseurs
    = from u in BDD.personnel
      where u.programmes.Count > 1000
      select u ;
```

Il s'agit d'une approche non standard du fait qu'il s'agit d'outils réservés au Framework de Microsoft, toutefois leurs utilisations améliorent grandement la productivité du développeur.

Ajout des options de configurations

Une fois la base de données créée, l'interface et les traitements ajoutés, l'application devient exploitable (une démonstration est alors faite à la direction de RML).

Les options de configurations sont toutefois nécessaires. En effet, si en soit l'ensemble des paramètres du projet peuvent être modifiés directement dans le code source, il est préférable de laisser un accès à certaines options pour l'utilisateur de l'application.

D'une part pour saisir les informations spécifiques au poste sur lequel l'instance de Corrinos s'exécute (type de machine, le répertoire local préféré pour la récupération des programmes), mais également pour qu'elle soit adaptable toutes modifications faites sur le réseau (changement du serveur, de mot de passe)...

L'ensemble de ces informations de configuration est éditable sur la barre de menu, et conservé dans un fichier texte *Corrinos.ini* de la manière suivante :

```
NomMachine=Fcn42;  
userLevel=0;  
PathLocal=D:\programmes\  
TypePoste=FCN;  
...
```

Intégration Corrinos dans Arrakys

L'ajout des informations de Corrinos dans le programme de gestion de Dune permettra d'une part au responsable de validation des commandes de mieux anticiper le temps nécessaire à la création d'une pièce mais également pour le chef d'atelier de mieux répartir la charge de travail en voyant à l'avance quels OF auront besoin d'une conception de programme.

Elle prend la forme d'une petite icône en bout de ligne et sera définie à partir des informations de la base de données de Corrinos, toujours en fonction de la référence plan attachée à l'OF / commande.

Cette partie du projet sera toutefois réalisée exclusivement par mon maitre d'apprentissage car elle touche à l'application de gestion réalisée en vb6.

Ajout du panneau de recherche

Moins primordiale que le panneau standard des encours montré plus haut, son utilisation est optionnelle (située vers l'étape de sélection de programmes existants), elle permet aux opérateurs de chercher directement un lot de programmes dans la base de données à partir de différents critères, le panneau standard se basant uniquement sur les références de plan des of en cours. C'est très utile si l'opérateur veut se baser sur un ancien programme pour un tout nouveau plan par exemple.

Nom	Lancement	Auteur	Référence Plan	Post
7 Sol1-Corrigée	13/06/2011	Aucun	Z000319641+EM...	Tour
6 Solution1	13/06/2011	Aucun	Z000319641+EM...	Tour

Solution1 [28464/11]
Auteur: **Aucun** Copier la sélection en local: >>

- programme1
- programme2
- programme3
- programme4
- programme5
- programme6

Formulaire de recherche multi-critères. Liste de solutions en réponse à la recherche. Equivalent de l'étape de 2 pour la copie en local.

On notera que l'utilisation de l'Entity Framework améliore grandement l'aisance à produire une RMC (Recherche Multi Critères) en peu de lignes.

En effet, si dans une utilisation standard de SQL on aurait été forcé de construire la requête morceau par morceau en fonction du remplissage des champs pendant l'exécution, avec l'Entity Framework on effectue simplement un filtrage successif de requête LINQ, la requête SQL est alors automatiquement générée lors de l'accès aux données en comportant en une fois tous les filtres précédemment appliqués dans le traitement.

Problèmes rencontrés

Le transfert des programmes directement vers les machines

Fonctionnalité très intéressante car elle élimine l'étape où l'opérateur doit lui-même copier les programmes récupérés depuis Corrinos vers la machine outils en passant par des outils propriétaires.

Toutefois comme énoncé dans la partie infrastructure ces transferts sont soumis à un protocole générique spécifique à chaque système embarqué sur les machines.

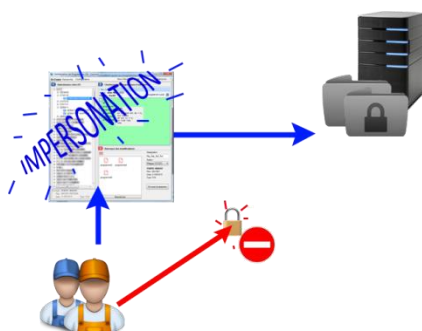
Il faut donc agir au cas par cas, en commençant par les machines les plus répandues. Pour RML il s'agit des machines-outils Hermle utilisant le terminal TNC 530. Après recherche, je remarque que le protocole d'échange se nomme LSV2, qu'il appartient à la société allemande Adontec/Supercom, et qu'aucune documentation sur son utilisation n'est disponible. Le SDK est vendu à hauteur de 500€ en ligne, il s'agit d'un surcoût trop haut pour être accepté. En alternative, je me suis rabattu sur l'utilisation en fond de tâche d'un logiciel fourni avec les machines. Une approche moins propre mais gratuite.

Restriction de l'accès au répertoire de centralisation

Le choix pris de centraliser les programmes-opérateurs sur le système de fichier du serveur en « partagé » a ses limites. En effet il est donc accessible à tous les utilisateurs des postes atelier, et toute personne est en mesure de détruire l'ensemble des fichiers inscrits. Pour pallier à ce problème il m'est demandé de limiter l'accès au répertoire de centralisation uniquement aux administrateurs du réseau.

Malheureusement, le client Corrinos aurait lui aussi besoin du statut d'administrateur réseau pour poursuivre l'exploration et la copie des programmes sur le partage, hors l'application hérite de l'utilisateur de la session à partir de laquelle il a été lancé. En clair, pour que Corrinos continue d'accéder au serveur des programmes, il faut également passer les opérateurs en administrateurs, ce qui ne résout pas le problème.

Après recherches, il est tout à fait possible de donner à une application un statut d'utilisateur différent de la session sur laquelle elle a été lancée, on appelle ceci l'impersonation. Je me base sur un code disponible sur codeproject.com pour créer un contexte d'impersonation lors des accès au serveur.



De ce fait, Corrinos accède au partage en changeant son identité temporairement, tandis que l'utilisateur de l'application conserve ses droits d'origines et ne peut y accéder de lui-même.

Toutefois ce changement est bien évidemment soumis à l'utilisation d'un identifiant et d'un mot de passe administrateur par l'application.

Sécurité relative à la conservation de l'identifiant

Comme toute propriété vouée à changer dans le temps (installation d'un nouveau serveur ?), les identifiants nécessaires à Corrinos pour utiliser l'impersonation doivent être saisis et mémorisés.

Toutefois le fichier Corrinos.ini est accessible à l'utilisateur, et il ne faut pas qu'il soit en mesure de lire les identifiants administrateurs. Pour cela je choisis d'encoder le mot de passe avec un algorithme de cryptage* XOR trouvé sur stackoverflow.fr.

C'est un algorithme rapide et facile à implémenter qui se base sur une clé secrète (qui sera définie dans le code source et inaccessible de l'utilisateur) afin de crypter et de décrypter des textes.

Principe de cryptage : Il s'agit à travers un algorithme mathématique (nécessitant parfois une clé) de transformer un texte donné lisible en un texte illisible, tout en conservant la possibilité de retrouver le texte source (contrairement au « hashage »).

Test

Fiabilité du réseau wifi de l'atelier

Le wifi est assez rarement conseillé en milieu professionnel, que ce soit pour sa fiabilité concernant la sécurité (plus facilement mis sur écoute que les liaisons filaires) mais également concernant la qualité de transmission (il est facilement soumis à de lourdes interférences). Dans notre cas, le point d'accès sera assuré par un routeur wifi dit industriel (transmission amplifiée) qui utilisera le cryptage WPA2 (dit non-crackable).

Pour s'assurer de la tenue de la transmission vis à vis des nombreuses sources de perturbations électromagnétiques de l'atelier, j'utilise mon ordinateur portable en exécutant un ping* continu sur le routeur wifi tout en me déplaçant partout dans l'atelier aux heures intensives d'utilisation des machines (afin d'identifier les potentielles sources de perturbations).

*Ping : requête standardisée, permettant de déterminer le temps (exprimé généralement en millisecondes) nécessaire à un échange entre deux éléments d'un réseau.

Le routeur jouera ici très bien son rôle, le délai du ping se limitant à quelques dizaine de ms.

Utilisation de l'application

Les tests comprennent également l'utilisation de Corrinos en environnement de production. On y vérifie en premier le lieu, le succès de la procédure de récupération et d'envoi de programmes-opérateur de différentes tailles, mais également la vitesse d'exécution des échanges. On en profitera aussi pour vérifier la compatibilité de communication entre les nouveaux pc servantes et les machines depuis le port RS232 avec les outils de transfert de programme tiers.

Aucun souci ne sera relevé pendant ces tests.

Le déploiement

Je n'avais, pour l'heure, généré aucun installeur avec Visual Studio or, après recherche, outre les projets spécialisés dans la conception de fichier d'installation, Visual Studio fourni un outil automatique qui, en plus de constituer automatiquement un installeur, intègre un système de mise à jour automatique de l'application.

Cette technologie nommée « ClickOnce » permettra donc, après une brève installation accessible à n'importe quel utilisateur, une mise en place des correctifs sans intervention de ma part dans l'atelier sur chaque poste servante.

Son fonctionnement est assez simple puisque il vérifie juste l'existence d'un installeur mise à jour sur une destination définie (dans notre cas, ce sera le serveur de la société RML).

La formation

Requise lors de la mise en place de toute nouvelle application spécifique, la formation est d'autant plus nécessaire dans le cadre de notre projet car elle est condition d'un remboursement partiel du projet par un organisme de formation (l'ADEFIM).

Elle sera organisée par groupe de 5/6 opérateurs. Ce nombre permet d'une part de se concentrer sur les questions de chacun et d'autre part de conserver une certaine tranquillité des employés qui pourront en plus profiter des questions des autres.

A l'issue de la formation, une documentation utilisateur (résumant les points expliqués) est remise pour garder en mémoire l'essentiel de l'utilisation du produit.

Améliorations possibles

Augmenter le nombre machines-outils compatibles avec le transfert de Corrinos

C'est un travail long et fastidieux qui peut être spécifique à tout modèle de machine. Toutefois son implémentation pourrait faire économiser de précieuses secondes et faciliter l'emploi de l'application.

Néanmoins les difficultés sont nombreuses (pas de machine de tests, complexité du protocole d'échange, protocole propriétaire non révélé).



Gestion de la zapette

Elle est utilisée pour la gestion des temps des employés afin d'estimer le temps nécessaire à l'usinage de telle ou telle pièce, ou pour un aperçu temps réel de la charge dans l'atelier.

Son fonctionnement qui consiste à zapper un code barre représentant l'opérateur puis un code barre sur l'OF papier pourrait être directement intégrée dans Corrinos afin entre-autres d'automatiser la sélection de l'of dans l'arbre de l'étape 1.

Intégration d'un éditeur de programme

Si Corrinos se charge pour l'instant uniquement de la récupération et de l'envoi des programmes, il pourrait être intéressant d'y inclure un éditeur de programme afin de donner la possibilité aux opérateurs d'effectuer des modifications mineurs sur le poste sans avoir à changer de programmes. En outre la conception d'un tel éditeur intégré inclurait les outils propres à la création de programmes pour machine outils comme par exemple une colorisation de la syntaxe.

Conclusion

Ce projet aura été pour moi le premier en relation avec client. Cela m'a permis pour la première fois d'exercer au mieux mon travail d'analyse afin de mieux cerner ce domaine que je ne connaissais pas du tout, de faire face à des contraintes techniques coriaces et réfléchir aux solutions possibles, ainsi que d'apprendre à mieux organiser mon code dans le cadre d'un gros projets.

De plus cela aura été l'occasion pour moi de poursuivre ma découverte des outils de développement actuels tel que Linq2Entity pour l'accès aux données, ClickOnce pour le déploiement et la mise à jour des applications, et Team Foundation Server pour la gestion du code source.

Des acquis je ne manquerai pas à mon tour de partager, d'exploiter et d'enrichir avec mes futurs projets pour la société Dune.