LES OUTILS LOGICIELS

Document de Claude Hernoux et Jean-Yves Marjou
avec la collaboration de Roger Gouriou, Jean-Yves Meuric, Michel Ruvoën

L'objectif de cette contribution est de décrire l'évolution des outils qui ont permis de passer des instructions écrites par les développeurs de programmes logiciels aux informations enregistrées dans les mémoires du commutateur téléphonique E10 et qui ont permis aussi l'informatisation de la documentation ou la gestion du matériel.

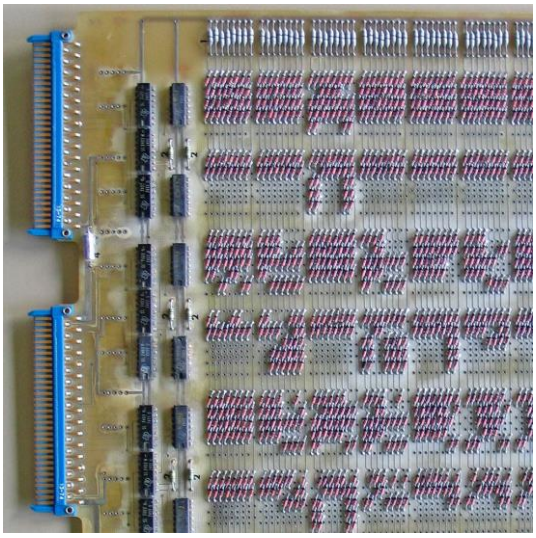
Pour simplifier la présentation des outils logiciels, la contribution range les évolutions dans un nombre réduit de grandes étapes assimilables par un maximum de lecteurs.

1 – Prémices

Avant 1976, la production du logiciel du Commutateur E10 ne fait pas appel, sauf quelques cas, à un calculateur central.

On distingue schématiquement :

- Le logiciel du CTI : le développeur perfore les instructions de son programme sur une machine à perforer les cartes, puis fait la compilation et l'édition de liens du programme résultant directement sur la machine cible 10010 ou MITRA 15 puis MITRA225; la hantise du développeur est de voir chuter les cartes perforées de son bac ou même de permuter des cartes qui feront échouer la compilation ; le logiciel exécutable sort sur ruban perforé puis sur bande magnétique (la galette) et est chargé et conservé sur le disque du CTI.



- Le logiciel des Organes de Commande et des Unités de Raccordement : le développeur code les instructions de son programme directement en instructions de la machine cible. Le programme se présente sous forme de diodes soudées sur une carte « Mémoire-Programme » dans une matrice lignes/colonnes. Le développeur doit avoir une connaissance très fine du fonctionnement de la machine. Les corrections font appel au fer à souder et au multimètre pour déceler les diodes à l'envers ou les courts-circuits. La machine ne nécessite pas de chargement de

logiciel ; elle est immédiatement opérationnelle dès la mise sous tension (plus tard, la technologie évoluant, les diodes seront remplacées par des mémoires PROM et REPROM).

- Les cas particuliers : par exemple le calcul des filtres numériques (reconnaissance des tonalités et de la signalisation multifréquence entre les commutateurs) fait appel à un logiciel écrit en langage FORTRAN. Les données sont portées par des cartes perforées et traitées la nuit sur le calculateur du CNET. Plus tard le PDP11 de l'équipe de test sera également utilisé.

Différents outils et évolutions apparaissent progressivement. Ainsi :

- En 1974, l'équipe machines de test développe un traducteur d'instructions de l' ELS (le processeur des Unités de Raccordement). Ce programme est écrit en Fortran et s'exécute sur le PDP11 qui pilote également le testeur de cartes logiques Oracle. Un ruban perforé est généré sur PDP11 avec la description des cartes à diodes MPD2. Ce ruban est ensuite utilisé comme données d'entrée :
  - du programme de test des cartes MPD (sur Oracle ou Becmad)
  - d'un programme qui génère les données de traçage des « DRM » ; ce DRM faisait partie du dossier élaboré par le bureau d'études
  - de la machine d'aide à l'insertion des diodes qui était utilisée par la Direction Industrielle. Il s'agissait d'une table qui venait éclairer tour à tour les emplacements à équiper d'une diode. Cette insertion était manuelle.

- A partir de 1974, l'ELS est utilisé également comme processeur pour les organes de commande dans CITEDIS (commutateur privé) puis dans E10-B. Le volume du logiciel étant nettement plus conséquent, il est développé un langage d'assemblage pour les fonctions de service (une instruction assembleur pour une instruction ELS) et un macro-langage pour le traitement d'appel (une macro-instruction générant plusieurs instructions ELS). Le développeur peut s'affranchir dans une certaine mesure de la connaissance fine de l'architecture de la machine et la lecture des programmes est facilitée par l'utilisation de symboles. Un assembleur et un macro-assembleur ELS sont développés par l'équipe de SLE-Citerel à Boulogne. Ces programmes s'exécutent sur un IRIS80 situé à Boulogne. L'accès à cette machine se fait via un « terminal lourd » muni d'un lecteur de cartes perforées et d'un perforateur de ruban.

- Progressivement les matrices à diodes sont remplacées par des mémoires REPROM et pendant la mise au point du logiciel en maquette par des mémoires vives (outil CHARME, composé des cartes ANG et LIC). La chaine de production se rapproche de celle du CTI. Le développeur ELS perfore ses cartes, lance un assemblage au CDC (Centre de Calcul) et récupère bande et listing. Mais on a encore gardé l'esprit "diodes", ce qui fait qu'il n'est pas rare de voir certains faire quelques « patchs » sur la bande avec la pointe d'un compas ou d'un fer à souder - crime de lèse-majesté : ceci ne se fait pas au CTI !!!.

- A partir de 1978, les Unités de Raccordement tournent sur des microprocesseurs du commerce et le développement du logiciel se fait sur une machine INTELLECT qui produit un fichier chargeable sur disquette.

- Plus tard, nous avons connu le support K7 Texas puis les disquettes. Et enfin la révolution avec l'avènement des PC, après une courte apparition des MDS en maquette (MDS d'INTEL, permettant de faire bien plus que les développements pour les processeurs du même nom!!!)

Le logiciel est en assembleur pour les organes de commande et les Unités de raccordement et en CPL1 puis en CHILL pour le CTI à l'occasion du marché chinois en 1986.

Dans cette période, le développeur se déplace physiquement près de la machine à perforer et de la machine cible pour le test de son programme.

Les outils de mise au point ont évolué depuis les fameux pupitres jusqu'aux MDS et PC ; sans compter les simulateurs d'environnement.
A propos des pupitres, qui ne se souvient du fameux pupitre ELS ? Le soir, quand la lumière baissait, ces pupitres faisaient l'admiration des visiteurs, avec leurs centaines de diodes LED (rouges bien sûr, c'était la seule couleur existante), qui n'arrêtaient pas de clignoter dans tous les sens. A cette époque, les traces n'existaient pas encore, il fallait se satisfaire du "Traceur" et du "Codeur d'arrêt" offerts par le pupitre. Mais un metteur au point expérimenté, pouvait se rendre compte de la bonne marche de son programme, uniquement en voyant clignoter les LED du pupitre ; cette facilité disparaît plus tard lors de l'introduction des tests en ligne de parité de mémoire vive (RAM), tests qui tournent sans arrêt.

Vers 1976, le site de Lannion est doté d'un calculateur central IRIS80 placé au rez-de-chaussée du bâtiment 2. La climatisation de la salle fait appel à l'eau de la « piscine » (le bassin situé au niveau de l'entrée historique de la SLE entre les bâtiments 1 et 2). De ce fait la température de l'eau de la piscine s'élève, laissant apparaitre des algues vertes (tiens déjà !).

Le développeur met ses cartes dans sa case près du calculateur, un opérateur passe son travail (job) et le développeur récupère plus tard le listing et son programme sur bande magnétique.

Plus tard, vers 1980, chaque développeur dispose dans son bureau d'un terminal sans intelligence (un écran et un clavier), une console VM/CMS qui lui permet d'écrire dans un fichier informatique.

A cette époque, la multiplicité des clients et des variantes de logiciel impose de gérer l'évolution :
- de la structure du réseau (abonnés isolés, centre d'affaire),
- des besoins (facilités)
- et l'intégration du traitement des données

Par ailleurs, l'évolution des technologies (intégration des composants, développement du hardware, séparation firmware / logiciel, capacités mémoire, performances des processeurs) d'une part et l'évolution de la normalisation des télécommunications et la standardisation des protocoles d'autre part vont conduire en parallèle à une évolution de l'architecture matérielle et à une explosion du volume de logiciel dont les conséquences directes seront une croissance du nombre de sites de développements et du nombre de développeurs ainsi qu'une augmentation du nombre de versions du logiciel à maintenir et traiter simultanément.

Ces contraintes donnent naissance à deux spécialités du génie logiciel :

- la gestion de configuration chargée de définir l'enchaînement et le contenu fonctionnel des versions
- les ateliers de génie logiciel chargés de gérer les modules logiciels

Nous allons nous intéresser plus particulièrement aux ateliers de génie logiciel.

2 – L'atelier de génie de Logiciel SDL 1976-1987

Le développement de l'informatique a essentiellement commencé par la mise au point de machines offrant des capacités de calcul et de mémoire de plus en plus importantes (loi de Moore).
Mais le développement du logiciel est resté longtemps archaïque, sans offre d'atelier intégré sur le marché.

Les premiers balbutiements de gestion de logiciel ont été réalisés par l'apparition de la notion d'update, la saisie étant alors réalisée à partir de cartes qui avaient la fâcheuse tendance de bourrer dans les lecteurs ou de se mélanger pendant les transports et manipulations.
L'update permettait d'identifier les cartes modifiées et de ne manipuler que celles-ci.
Avec l'update, la notion de patch est apparue, les évolutions pouvant être réintégrées dans le source, c'est-à-dire donnant la possibilité de recréer un nouvel ensemble de cartes mis à jour.

Le passage de la saisie sur cartes perforées à la saisie par des terminaux, le développement de bibliothèques de programme et l'arrivée de base de données ont permis des avancées dans le domaine de la gestion et ont abouti à la création de réels ateliers de génie logiciel.

Compte tenu du manque d'offre, des solutions internes ont été développées.

Un premier atelier de génie de logiciel est développé en interne et baptisé Système de Développement de Logiciel (SDL) ; il permet de nommer les modules logiciels, de leur attribuer une version et de les rattacher à une arborescence pour qu'ils utilisent les « inclus » et les outils de génération de code spécifiques à leur version.
Les messages échangés entre machines sur les bus de communication du commutateur, jusqu'à cette date dessinés graphiquement sur du papier, sont alors codés comme des « inclus ».

Premier atelier CSE, début du CSN :
Ce premier atelier a été développé sur la base de bibliothèques de programmes, à chaque version est associée une bibliothèque qui évolue en numéro d'édition.
La correction d'anomalies et l'introduction d'évolutions impliquent de gérer en parallèle plusieurs bibliothèques.

La documentation est écrite sous DCF.

Le matériel est géré sous GP.

3 – L'atelier de génie logiciel VM/SE

Le calculateur central devient de plus en plus gros pour répondre aux besoins des développeurs. L'IRIS80 laisse place à un IBM dont le volume de mémoire de travail (RAM), le volume disque et la puissance de calcul (puissance UC) ne cessent de croître.

Le logiciel de développement des Unités de raccordement est dorénavant écrit en PLM (le langage de haut niveau adapté aux microprocesseurs).

Dans cette période, le logiciel du commutateur E10 est majoritairement en langage évolué ou de haut niveau (langage CHILL, PLM puis C …), le logiciel est compilé et mis au point sur machine hôte, à savoir le calculateur central.

La plupart des travaux sont possibles depuis le bureau du développeur à partir de son terminal VM/CMS.

L'environnement de développement, c'est-à-dire la gestion de configuration logicielle, se fait sous VM/SE
VM/SE (Virtual Machine Software Engineering 1986).
Nota : VM est un OS (Operating System) IBM qui affecte à chaque utilisateur un espace mémoire, un espace disque et des ressources UC.
Le premier véritable atelier de génie logiciel apparaît avec l'abandon des patchs techniques qui s'avèrent non adaptés lorsque le cycle de développement s'accélère : le nombre de patchs à intégrer pour la version suivante devient important et l'effort nécessaire pour obtenir un produit stable lors de l'intégration des patchs devient prohibitif.

La technique de modification du code source et la refabrication systématique des produits est adoptée.

La maîtrise du logiciel ne signifie pas simplement maîtriser l'évolution des sources, il est aussi nécessaire d'y intégrer les outils qui évoluent également du fait du développement de l'informatique qui introduit les compilateurs, éditeurs de liens,… nécessaires.
Ceci provoque l'abandon des outils « maison » et induit, du fait de l'obsolescence plus rapide de ces produits, des évolutions à prendre en compte.
Certaines versions de ces outils introduisent des incompatibilités avec les produits fabriqués à l'aide de la version précédente.

Les outils du commerce ayant une vocation universelle comportent de nombreuses options allant du format du listing aux options de génération du code (adressage absolu, relatif , binaire plus ou moins optimisé, plus ou moins compact,….).
Laisser la maitrise de ces options à chaque développeur est une source potentielle de problèmes, les différentes incompatibilités pouvant être découvertes en phase de fabrication du logiciel, en phase de tests unitaires, en phase d'intégration, en phase de validation ou même sur des produits en service.
Le coût de la correction de ces incompatibilités peut donc s'avérer très important.

La simple taille d'un listing peut varier de 1 à 10 en fonction des options choisies, ceci peut induire des volumes considérables d'espace disque, surtout lorsque le nombre d'objets se compte par milliers.

Pour éviter ces sources d'aléas, VM/SE intègre une notion de procédure de fabrication qui permet à partir d'un objet A de générer un objet B en utilisant un outil (compilateur, éditeur, linker, …) avec un ensemble d'attributs prédéfinis.

Le cycle de vie des logiciels comporte plusieurs étapes: spécification, codage, fabrication, tests unitaires, tests d'intégration, de validation. Le logiciel d'une étape fonctionnelle (palier) est souvent développé sous forme de plusieurs lots successifs jusqu'à complétude des fonctions de cette étape. Compte tenu des volumes logiciels, le nombre de développeurs est important (il a été au maximum de 400, à confirmer) et plusieurs développeurs peuvent intervenir sur un même logiciel. La production du logiciel doit donc gérer des états de partage et d'avancement.

VM/SE intègre donc un attribut d'état des objets qui va de la propriété d'un objet associée à un ou plusieurs individus au partage des objets.

Le développeur manipule des sources qui sont compilées, regroupées en modules, eux-mêmes regroupés en exécutables, puis en archives puis en logiciel chargeable sur la machine cible.

D'une version à une autre, le nombre de sources qui sont modifiées pour cause d'évolution fonctionnelle ou correction d'anomalies est variable.
Compte tenu de la méthode utilisée la version n+1 comportera x logiciels issus des versions antérieures et y issus de la version n+1.
La notion de version est associée à la gestion de configuration qui définit en fonction des évolutions fonctionnelles et des lots de correction le contenu de ces versions.
En termes VM/SE, la gestion de configuration est associée aux relations de domaines qui permettent d'hériter de l'ensemble des modules issus des versions antérieures.

La gestion de configuration et la maîtrise du produit nécessitent de connaître la liste exhaustive des composants logiciels du produit.

Pour satisfaire cette exigence, VM/SE est construit autour d'une base de données relationnelle.
Chaque objet est identifié par :

- son nom
- son genre (associé à la procédure de fabrication)
- sa version (liée à la gestion de configuration)
- son édition
- son itération
- son état (créé, validé, livré, intégré, validé, archivé).

Les objets sont : les sources, les binaires, les exécutables, les archives, les chargeables, les procédures de fabrication.
Un objet peut avoir des attributs (listing, ……) associés à des genres secondaires.

La base de données relationnelle permet de connaître tous les composants d'un objet fabriqué via une procédure de fabrication. En terme VM/SE, il s'agit de la liste de dépendance d'un produit depuis l'objet hiérarchique le plus complexe jusqu'au plus simple : le code source.
VM/SE intègre également une notion de dépendance ascendante qui permet d'identifier tous les objets qui comportent un objet donné.
De même, il est possible d'identifier tous les objets fabriqués à l'aide d'une procédure de fabrication.
La construction de ces dépendances s'appuie sur les relations de domaine.

Ces facilités permettent donc à partir de l'évolution d'un source d'identifier tous les objets concernés et le cas échéant de les refabriquer en maîtrisant leur composition.

VM/SE intègre également des procédures de livraison qui permettent de partager des sous-ensembles du produit.
Cette facilité est utilisée pour réaliser du développement multi-site, l'inconvénient est que chaque site doit posséder un IBM tournant sous le système VM.

Enfin VM/SE intègre des facilités d'archivage et restauration qui permettent le retrait et le rechargement d'anciennes versions, ce qui optimise la gestion des espaces disques.

Limites de VM/SE

- VM n'offre pas d'interfaces graphiques, ces interfaces s'avèrent utiles et nécessaires en particulier pour les outils de spécification, de tests.
- Les outils de spécification et d'analyse objets ne sont pas supportés.
- Le raccordement de stations de compilation est laborieux et les protocoles de gestion de ces stations déportées sont peu évolués.
- Chaque site de développement doit posséder un système VM.

4 – L'AGL décentralisé

Pour faire face à l'accroissement du besoin en ressources informatiques et aussi pour permettre aux développeurs des Centres Techniques à l'Export (CTE) de produire leur logiciels, le calculateur central est réduit et complété par des serveurs de développement. Leur nombre prolifère: MacroMR commun à Lannion partagé avec le CTE Inde, MacroMR applications export à Orvault partagé avec le CTE Pakistan, UTC à Nantes partagée avec la Roumanie, NA à Lannion partagée avec Orvault et avec la RSA, Serveur des Essais de validation,…
Les terminaux des développeurs sont des Workstations puis des PC standards moins onéreux et plus équipés en outils informatiques puisque le langage C se généralise pour le logiciel.
L'architecture matérielle des ressources informatiques ne résout pas tous les cas d'utilisation car les conduits informatiques entre la France et les CTE ont un faible débit, aussi les développeurs des CTE ne travaillent pas directement dans le serveur cible mais sur une copie locale dans leur site. Leur production est ensuite rapatriée dans le serveur dédié en France qui est utilisé comme source pour les livraisons.

Les logiciels de la NA (Nouvelle Architecture du Traitement d'Appel) sont écrits en LDS, langage graphique, qui est transformé en instructions par l'outil informatique GEODE pour donner du LDS PR, transformé à son tour par l'outil informatique SOLANGE pour donner un source en langage C.
L'environnement de développement, c'est-à-dire la gestion de configuration logicielle se fait sous BENCHCOM.

Pendant longtemps, jusqu'en 2000, les livraisons conduisent le développeur à porter son logiciel sur une bande magnétique jusqu'au CTI ou à l'OM qui le charge dans le commutateur.

La documentation est écrite en DCF et centralisée sous VIDOC.

Pour pallier ces contraintes et faire face au nombre croissant de sites de développement (Lannion, Nantes, Vélizy, Afrique du Sud, Roumanie, Inde, Vietnam), la création d'un nouvel atelier de génie logiciel fonctionnant sous Unix et sur des stations de travail est lancée.

Les avantages de cet atelier sont :

- La prise en compte du mode graphique
- La disponibilité des ressources UC, disque et mémoire sur chaque poste de travail
- La disponibilité des outils sous Unix
- La facilité d'équipements de nouveaux centres de développement qui ne nécessitent qu'un serveur et quelques stations de travail.
- La prise en compte du développement des réseaux.

Le système choisi est le système Unix AIX d'IBM qui fonctionne sur des stations RS6000.

Le nouvel atelier de génie logiciel se nomme X/SE (1995 à consolider)

X/SE comporte les mêmes fonctionnalités que VM/SE, bénéficie des avantages du « down-sizing » qui permet de s'adapter facilement aux évolutions des équipes de développement et aux besoins en espace disque et puissance.

Faute de crédits, X/SE s'avèrera être un simple portage de VM/SE du monde VM au monde UNIX. X/SE ne bénéficiera pas de toutes les capacités offertes par UNIX, ni de celles offertes par l'évolution des bases de données relationnelles

Les inconvénients de X/SE sont :

- Le rythme d'évolution des matériels, des outils
- La gestion du parc des stations de travail, le coût de ces dernières.

Suite à un choix basé sur des questions de coûts des stations de travail développeurs, celles-ci seront abandonnées et remplacées par des PC fonctionnant sous Windows NT, puis XP.
Les PC sous Windows ne supportant pas tous les outils, des stations UNIX dédiées seront conservées.

Faute de financement et de volonté et du fait de l'évolution d'Alcatel, X/SE qui sera resté un produit maison CIT, sera abandonné au profit d'un outil du commerce ClearCase.

Malgré le nombre d'objets (plusieurs millions), le volume et le nombre de relations de la base de données, les espaces disques VM/SE et X/SE ont été des outils robustes et fiables qui ont supporté sans difficulté les évolutions de système et l'augmentation du nombre de développeurs.

5 – L'atelier de génie de Logiciel Clearcase

La première activité du E10 à utiliser Clearcase est l'écriture des essais automatiques.

Le langage JAVA ou UML est utilisé pour le DHA et pour le logiciel SMB de gestion des serveurs INCS2.
L'environnement de développement, c'est-à-dire la gestion de configuration logicielle se fait sous Clearcase.

Dans cette période, un effort est fait pour introduire du langage orienté objet dans le logiciel du commutateur E10.

6 – Conclusion

Voilà dans quel environnement sont nées les 5 millions de ligne source (hors commentaires) qui rendaient les commutateurs E10 aptes à servir les besoins téléphoniques des abonnés dans différents pays du monde en mi 2004.
Les serveurs de développement étaient un premier pas vers les serveurs HTTP du Web du monde Internet où les langages PHP, bases de données MySQL ont pris le relais des premiers langages évolués.

Glossaire :

10010 : calculateur du CTI pour Platon et E10 niveau 4
AGL : Atelier Génie Logiciel
AIX : Advanced Interactive eXecutive (Système d'exploitation de type UNIX d'IBM)
BECMAD : Banc d'Essais des Cartes Mémoire A Diodes
CAO : Conception Assistée par Ordinateur
CCITT : Comité Consultatif International des Télécommunications
CDC : Centre de Calcul
CHARME : CHARgeur de MEmoires
CHILL : langage de développement de logiciel préconisé par le CCITT
CII : Compagnie Internationale d'Informatique
CIT : Compagnie Industrielle des Télécommunications
CITEDIS : commutateur privé de type E10
CITEREL : CIT-Ericsson-Electronique
CLEARCASE : Gestion de Configuration de logiciels
CMS : Console Monitor System
CNET : Centre National d'Etudes des Télécommunications
CP80 : carte processeur équipée du processeur Intel 8080
CPL1 : langage pour écrire un programme informatique
CSE : Concentrateur Satellite Electronique
CSN : Centre Satellite Numérique
CTE : Centre Technique Export
CTI : Centre de Traitement des Informations
DCF : Document Composition Facility (Outil IBM de présentation de documents)
DH : Division Hardware, ancienne appellation de DMM
DHA : Data Handling Application (successeur du TR)
DMM Division de développement des matériels
DRI : marque de disque magnétique à cartouche
DRM : D (code attribué par la SAS au constructeur SLE) R (format) M(contenu) du plan d'équipement
DRT Direction Régionale des Télécoms
DSF : Dispositif de Sauvegarde de Fichiers
DSI : Division Systèmes Informatiques
E10 : Commutateur Electronique Temporel
E10A : Commutateur Electronique Temporel de 1971 à 1981
E10B : Commutateur Electronique Temporel de 1981 à 2001
E10S Commutateur générique initial avec des GUT pour raccordements d'abonnés
ECH : Echangeur (logiciel sans macro ou micro programme)
ELS : Equipement de Logique Standard
FORTRAN : langage évolué de programmation
GP : Gestion Produit
http : HyperText Transfer Protocol
IBM : International Business Machine
INTEL : Fabriquant de semi-conducteurs
INTELLECT : calculateur pour développement de logiciels
IRIS 80 : calculateur à usage de centre de calcul pour le développement de logiciels
JAVA : langage de programmation informatique orienté objet
LDS : Langage de codage sous forme graphique (SDL en Anglais)
LED : Light Emitting Diode (diode électroluminescente)
MACHPRO : Machine à reprogrammer des mémoires
MDS : Mini Disk System

MINEX : Pupitre de mise au point à base de PC
MITRA : calculateur sur 16 éléments binaires fabriqué par la SEMS
ML : Machine Logique
MLSM : ML Station Multiplex
MPD : Mémoire Programme à Diodes
MQ : Marqueur (et interconnexion de BUS)
MR : Multienregisteur (contrôle d'appels)
MySQL : Base de données relationnelle « freeware » (SQL : Structured Query Language)
NA : Nouvelle Architecture du Traitement d'Appels
OCB181 : Commutateur E10 mis en service dès 1981
OCB283 : Commutateur E10 de capacité >=2000 MIC
OM : Operation and Maintenance (CTI mono central de l'OCB283)
OMC83 : calculateur de CTI basé sur l'architecture A8300 mais en mécanique OCB283
OMPC : version du TI
ORACLE : Organe Réalisant Automatiquement le Contrôle Logique des Equipements
OS : Operating System
PASCAL : langage évolué de programmation
PATCH : modification apportée directement dans le code exécutable d'un logiciel sans nouvelle compilation
PC : Personal Computer
PDP11 : Programmed Data Processor, calculateur à 16 eb de la Société Digital Equipment
PDP8 : Programmed Data Processor, calculateur à 12 eb de la Société Digital Equipment
PHP : Hypertext Preprocessor
RAM : Random Access Memory (mémoire vive ou de travail)
REPROM : boitier mémoire reprogrammable
SAR-CLER : cartes mémoires équipées de mémoires reprogrammables
SDL : Système de développement de Logiciels (AGL)
SEMS : Société Européenne de Mini informatique et de Systèmes (dont le MITRA)
SLE : Société Lannionnaise d'Electronique
SMB : Station Multiprocesseurs Banalisée
TAP Logiciel de Traitement d'Appels
TX : Taxeur
URA : Unité de Raccordement d'Abonnés
UC : Unité centrale
UD50 : Unité Disque de 50 Moctets
UML : Unified Modeling Language (langage de modélisation graphique à base de pictogrammes)
UNIX : « Operating System » d'origine Bell Labs tournant sur pratiquement tous les types d'ordinateurs.
UR : Unité de Raccordement (d'abonnés ou de circuits)
URA : Unité de Raccordement d'Abonnés
UTC : Unité de Traitement Canal sémaphore
VIDOC : Visualisation de document
VM : Virtual Machine
VM/CMS : Virtual Machine / Console Monitor System
VM/SE : système de gestion de configuration de développement sur calculateur centralisé
WADS : Word Alcatel
WEB : toile d'araignée. Désigne généralement le « World Wide Web », ensemble des points d'accès HTTP disponibles sur Internet.
X/SE : système de gestion de configuration de développement sur sous UNIX