

LTR, Langage Temps Réel

Henri Habrias
`henri.habrias@univ-nantes.fr`

1 Introduction

A l'Arsenal d'Indret sur la rive droite de la Loire, le SBBD Socrate¹ et le langage LTR étaient utilisés. Mais nous n'avons pu retrouver le personnel qui collaborait à l'enseignant du département informatique de l'IUT de Nantes et qui les utilisait. Il existe peu de documentation disponible actuellement sur le langage LTR. Nous commençons à présenter ce que nous avons trouvé sur la Toile. Puis nous reproduisons ce qu'a bien voulu nous adresser pour reproduction David Andreu qui fut enseignant-chercheur à l'Université Montpellier II avant de créer une entreprise. On y lira comment il a utilisé LTR pour son enseignement, son cours, des exemples de travaux pratiques et d'examens. Enfin nous fournissons les informations reçues la semaine précédant la journée scientifique de Nantes sur le langage LTR, J.-R Abrial et le Centre de Programmation de la Marine.

2 LTR sur la Toile

2.1 1968

Sur le site de la BNF on trouve :

- Description du langage, ECA automation, Saint-Cloud, 1968
- Description du moniteur opérationnel standard, ECA automation, Saint-Cloud, 1968
- Introduction générale au langage, ECA automation, Saint-Cloud, 1968
- Spécifications du langage temps réel, ECA automation, Saint-Cloud, 1968
- Description du compilateur, ECA automation, Saint-Cloud, 1968
- Description, rôle et mise en œuvre du précompilateur, ECA automation, Saint-Cloud, 1968

2.2 1980

Langage Temps-Réel

" (French for "real-time language") A French predecessor to Ada, LTR is Modula-like with a set of special-purpose real-time constructs based on an event model. It was mentioned in the reference below. ["An Overview of Ada", J.G.P. Barnes, Soft Prac& Exp 10 :851-887 (1980)]

LTR3

1. Je me souviens de la déception des informaticiens lorsqu'ils ont dû quitter Socrate (qui était devenu Clio) pour Oracle

Version three of LTR, by A. Parayre of Delegation Generale pour l'Armement, France. LTR3 was widely used by the French military and avionics companies.

["The LTR3 Reference Manual", A. Parayre, Delegation Generale pour l'Armement, France].

2.3 1985

"Les langages modula-2, LTR3 ou Ada sont des descendants directs de Pascal, qui fut à l'origine un langage pensé pour l'apprentissage de la programmation." Jean-Jacques Dupas, dossier : Un esprit curieux des sciences, HS Kiosque 85 : Blaise Pascal <https://tangente-mag.com/article.php?id=7235>

2.4 1986

Études sur l'approche base de données pour la gestion d'objets issus du développement de logiciel, Myung-Joon Kim, thèse sous la direction de Jean-Claude Derniame, 1986, Université Henri Poincaré Nancy 1. Faculté des sciences et techniques.

" Les travaux menés se divisent en trois étapes progressives :

1. Étude de la base de données relationnelle et intégration des applications de base de données relationnelle dans le langage LTR-3 à l'aide de la notion de type abstrait de données ;
2. Études approfondies sur le modèle entité-association et proposition d'une méthode de transformation du modèle entité-association au modèle relationnel ;
3. Première description d'un système de gestion d'objets basé sur le modèle entité-association avec la spécification de ses fonctions "

Dans cette thèse on apprend très peu sur LTR-3. Mais on a dans la bibliographie :

- Spécifications fonctionnelles de l'atelier LTR3, Centre d'Electronique de l'ARMement, décembre 1983
- LTR-3 Manuel Officiel de Référence, Compagnie d'Informatique Militaire Spatiale et Aéronautique, 1983

J'ai consulté le site des Anciens Cols Bleus et Pompons Rouges <https://www.anciens-cols-bleus.net/> où on apprend que "Les deux filiales THOMSON étaient CIMSA pour le matériel destiné aux militaires et SEMS pour le matos civil (années 70)."

- SEMS = Société Européenne de Mini-informatique et Systèmes.
- CIMSA = Compagnie d'Informatique Militaire Spatiale& Aéronautique.
- Notons que la Compagnie internationale pour l'informatique - Division militaire, spatiale et aéronautique (CII - DMSA) est devenue CIMSA en 1977

Le Mitra15, Machine Industrielle pour le Temps Réel et les Automatismes, est un mini ordinateur 16 bits temps réel de technologie TTL qui comporte une mémoire principale à tores de ferrite au lithium organisée en mots de 16 bits. Chaque carte mémoire est un bloc de 4096 mots de 16 bits et on peut utiliser jusqu'à 8 cartes mémoire. Il a été construit par CII (Compagnie Internationale pour l'Informatique) à plus de 7000 exemplaires de 1971 à 1985. Cet équipement

a fait partie des premières réalisations utilisant des calculateurs pour la conduite du système électrique EDF.

voir aussi : *Chronologie aéronautique et spatiale, civile et militaire 1939-2009*, Claude Carlier, Économica, 2009

2.5 Début des années 2000

"Mon premier travail consistait à porter un générateur de code LTR3 vers une machine cible ésotérique, au début des années 90. De mémoire, c'était effectivement un concurrent franco-français du langage Ada83. Le langage était du même style que Modula-2, c'est-à-dire du Pascal modulaire. Son gros intérêt était qu'il intégrait plusieurs primitives liées au temps-réel et à la gestion de la concurrence. En particulier, il avait déjà une variante des protected objects qui ne sont apparus que dans Ada 95. C'est chez ClearSy que j'ai eu ma dernière occasion de côtoyer LTR3 (portage d'un vieux compilateur SYSECA sur une machine plus récente) au début des années 2000. Sinon, notre interlocuteur côté DGA était le CELAR à Rennes qui chapeautait tous les développements liés à LTR3." Laurent Voisin, Systerel, 2025

note : "Protected objects are intended to provide a "lightweight" mechanism for mutual exclusion and data synchronization. You should use a task only when you need to introduce explicitly a new, concurrent thread of control . Protected objects offer a low overhead, efficient means to coordinate access to shared data. A protected type declaration is similar to a program unit and consists of both a specification and a body. The data to be protected must be declared in the specification, as well as the operations that can be used to manipulate this data. If some operations are only allowed conditionally, entries must be provided. Ada 95 rules require that entry barriers be evaluated at the end of procedure calls and entry calls on protected objects. Entry barriers should avoid referring to global variables so that the underlying assumptions of the state of the protected object are not violated. Protected procedures and entries should be used to change the state of a protected object. " <https://www.adaic.org/>

Thierry Lecomte (Clearsy) m'apprend que le nom du compilateur était Entreprise.

3 David Andreu à l'université de Montpellier II

A l'Université de Montpellier II, David Andreu faisait des Travaux Dirigés d'Informatique Industrielle avec le Langage Temps-Réel (LTR). Il nous a transmis son cours, travaux pratiques, sujets d'examens, dont il nous a autorisé la reproduction. David Andreu est Cofondateur, CTO et Directeur Général de la société Neurinnov <https://neurinnov.com/> dont le siège est à Montpellier. Il est un spécialiste de l'architecture des dispositifs médicaux implantables actifs (DMIA), domaine dans lequel il a remporté de nombreux prix. concevoir des solutions innovantes visant à pallier des déficiences fonctionnelles humaines, à la suite par exemple d'une lésion à la moelle épinière, grâce à la stimulation électrique fonctionnelle (SEF), permettant ainsi aux patients de retrouver de l'autonomie. David a publié 100 articles scientifiques et est auteur/titulaire de 9 brevets.

Nous le remercions vivement pour ce qu'il nous a adressé et autorisé à reproduire ici.

3.1 Introduction

L'enseignement du temps-réel (système d'exploitation et système temps-réel), dans une formation ingénieur électronique / robotique / informatique industrielle (public de 4ème année), était couplé à l'enseignement d'un formalisme de spécification des applications (ici, temps-réel) que sont les réseaux de Petri.

Sur un plan pédagogique : les 2 matières étaient enseignées en parallèle, TDs sur la partie modélisation, pas de TD sur la partie LTR (cours basé sur un exemple complet) et les TPs étaient organisés sous la forme d'un seul et même mini-projet couplant les 2 aspects (un volume d'environ 24h de mémoire) :

1. concevoir (architecture de l'application de commande TR, mécanismes IPC, configuration OS TR) et formaliser (modéliser le comportement des tâches et de leurs interactions)
2. analyser (logiciel Tina) et corriger si besoin (ex : recherche de deadlock)
3. mettre en œuvre (lang. C) sur un OS temps-réel « concret » (ce fut QNX au début, puis le micro-noyau FreeRTOS). L'étudiant n'apprend pas un OS particulier, il apprend à identifier son besoin et de lui-même il cherche dans la documentation de l'OS et comprend les primitives y satisfaisant. . . Ils sont ainsi capables de s'adapter à tous OS TR s'ils maîtrisent les concepts de base.
4. tester en pilotant un simulateur du système à contrôler (IT'S PLC de RealGames, interfacé avec une carte de contrôle externe), en observant aussi le comportement de l'application via un traceur (Perceptio).

Concernant le cours de LTR, après les généralités sur les systèmes TR, la partie LTR était enseignée à travers un exemple que nous déroulions durant le cours. L'exemple était complété et « complexifié » au fil du cours. La partie LTR du support nous servait donc surtout comme “description” des principes de bases du langage LTR (et ses instructions) à laquelle nous avions recours pour traiter l'exemple. En résumé, le temps-réel était enseigné par l'exemple, en s'attachant à expliquer le comportement de l'application au regard des mécanismes ou primitives mises en jeu et le fonctionnement de nature “événementielle” au sein de l'OS... ce qui de fait nous amenait à comprendre l'impact / le rôle de l'OS dans l'exécution de l'application.

Je vous mets dans ce dossier : Une diversité de sujets d'examens portant sur le LTR (pour les exemples qu'ils contiennent) Le support de cours fourni aux étudiants. . . (je ne suis pas sûr de la version). Un exemple de mini-projet (série de TPs)

4 et en quelques jours l'histoire a été écrite ...

Un ancien étudiant de Nantes, maintenant professeur à l'Université Paris Nanterre et chercheur au lip6, Pascat Poizat, m'a adressé le 16 mai 2025 un courriel où il me signalait la page 42 d'un rapport de l'Assemblée Nationale :

"Avis fait au nom de la commission de la défense nationale et des forces armées sur le projet de loi de finances pour 2018 (n°235), tome VII, Défenses, Equipement des forces-Dissuasion, par M. Jean-Charles Larso.

Pour l'amiral Christophe Prazuck, la DGA comme les armées doivent prendre en compte le décalage croissant entre le rythme de l'innovation industrielle – qui s'accélère – et la durée de vie des matériels, qui s'établit autour de 35 à 40 ans en moyenne. Auparavant, un même bateau recevait des équipements de deux générations de technologies – à titre d'exemple, les frégates F70 disposent d'un système de combat basé sur le langage informatique « temps réel » (LTR) développé pour le suivi des fusées du programme Apollo, avec deux « couches » de technologies plus modernes. Pour la génération suivante de bateaux, on peut estimer que chaque bâtiment recevra plutôt une dizaine de « couches de technologies ». Aussi leur conception doit-elle intégrer davantage d'évolutivité ; c'est d'ailleurs là l'un des objectifs du programme de transformation du système d'information des armées (SIA) appelé « SIA C2 » (commandement & contrôle).

J'ai obtenu la prompte réponse de monsieur Chrispothe Prazuk le lendemain.

5 Témoignage de Christophe Prazuk, ancien chef d'état-major de la Marine

Christophe Prazuk est actuellement directeur du nouvel institut de l'Océan de Sorbonne Université.

Pour répondre à vos interrogations, j'ai interrogé l'amiral Edouard Guillaud, ancien chef d'état-major des armées et acteur passionné du Centre de Programmation de la Marine (CPM). Il écrit " Il y a eu au moins deux versions de LTR utilisées (LTR1 qui était une transcription de la version US, puis LTR 2, française et plus puissante) et une en gestation (LTR3) à l'époque de la mise au point d'ADA. Le CPM s'intéressait aussi à l'époque à des langages récents tels que C, C+ et C++, en raisons des bouleversements apportés par les puces modernes. Je pense que les réponses seraient complètes si elles étaient posées à un ingénieur de l'armement en poste au CPM de la grande époque. En particulier à l'Ingénieur en Chef de l'Armement (ICA) Pascal Grojean, remarquable spécialiste visionnaire. " Vous pouvez contacter M. Pascal Grojean à l'adresse suivante en vous recommandant de l'amiral Edouard Guillaud. En espérant que ce nouveau contact vous permettra de parfaitement documenter le tournant informatique de la marine qui a changé ses instruments comme ses tactiques. Merci de me tenir informé de vos découvertes futures."

5.1 Témoignage de Pascal Grojean

Je suis prêt à répondre à vos questions, bien que n'ayant pas personnellement connu JR ABRIAL. C'était déjà presque un mythe à

cette époque !

Le langage LTR a été utilisé par le CPM, dans sa version v2, pour la réalisation industrielle de plusieurs systèmes embarqués dans les années 80 :

- SENIT 4 des corvettes anti sous marines (CASM) type Georges LEYGUES : (SENIT : Système d'Exploitation Navale de l'Information Tactique).

C'est le premier système de traitement de l'information tactique (fusionnant les données radars et autres senseurs & planification de l'emploi des armes) développé au CPM/ST pour la marine française conçu avec des technologies françaises. Le système est conçu avec une architecture centralisée basée sur la version militarisée de l'IRIS 80 développé par la CII. L'abandon de cette gamme lors de la fusion avec Honeywell Bull n'a pas permis de reprendre cette architecture pour les programmes ultérieurs.

La corvette Georges LEYGUES a été admise au service actif en 1979 et sera désarmée en 1997. Il y a eu ensuite 6 autres CASM, la dernière ayant été mise en service en 1990 et désarmée en... 2022.

- SENIT 6 des corvettes anti aérienne type CASSARD²

Vu l'arrêt des IRIS, le CPM/ST s'est lancé dès 1978/1980 dans la conception d'un système basé sur un réseau de "petits" calculateurs pour équiper les corvettes anti aériennes de type CASSARD.

Le système était réparti sur 6 calculateurs 15M/125X (version militarisée des MITRA) reliés par le réseau Bus Standard Marine d'Electronique Serge Dassault, un 7ème calculateur servant de spare pour permettre une redondance active. La table des pistes (cad la structure de données stockant les informations sur les mobiles détectés) était donc distribuée sur les différents calculateurs, chaque calculateur gérant une grande fonction opérationnelle (traitement des données radar, traitement guerre électronique, traitement des liaisons de données tactiques L14 et surtout L11 etc).³

Ce système est encore aujourd'hui (de mon point de vue) un chef d'œuvre de conception et de réalisation. J'ai dirigé le service qui s'occupait de ces SENIT dans les années 80 et je conserve encore pour ses concepteurs une admiration sans borne.

Il n'y a eu que deux CAA⁴, pour des raisons budgétaires. Le CASSARD est entré en service en 1985 et a été désarmée en

2. Jacques Cassard, né le 30 septembre 1679 à Nantes et mort le 21 janvier 1740 dans la forteresse de Ham, après quatre ans de détention, est un marin et un corsaire français des XVII^e et XVIII^e siècles. Il est moins connu que son ami René Duguay-Trouin et que son célèbre cousin Robert Surcouf. Le Lycée Professionnel Maritime situé en bord de l'Erdre à Nantes porte le nom de Jacques Cassard. Un monument à Jacques Cassard se trouve rue de l'Hermitage, avec vue sur la Loire et la ville.

3. https://fr.wikipedia.org/wiki/Liaison_11, <https://www.defense.gouv.fr/sites/default/files/cicde/20170613-NP-PIA-3.50-LDT-2017.pdf>

4. Corvette Anti-Aérienne, FAA, Frégate Anti-Aérienne

2019. Le Jean BART a été désarmé en 2021. A priori le SENIT 6 a duré "jusqu'au bout", complété lors de sa refonte par le module logiciel du SENIT 8 (voir ci-dessous) traitant la Liaison Tactique L16.

— Système ATL 2 Le CPM développait également le logiciel du système d'information du Breguet ATLANTIC⁵ de nouvelle génération (ATL 2) en sous-traitance d'Aviation Marcel Dassault, ce qui était assez cocasse, mais montrait bien la technicité du CPM/ST à l'époque.

— SENIT 8 PA Charles de Gaulle

Quand les études du Porte Aéronefs Charles de Gaulle ont démarré, sous ma responsabilité (et malgré ma jeunesse!), j'ai naturellement essayé d'utiliser la nouvelle version LTR v3, en lançant le développement d'un prototype SENIT en LTRv3.

Pour différentes raisons (retard du programme PA CdG, concurrence de ADA, apparition d'UNIX et de C++, etc), LTR v3 n'a pas été utilisé pour le développement industriel.

JR ABRIAL a-t-il participé aux travaux sur LTRv3 (chapeauté par l'ICA PARAYRE de la DRET) ? Je ne l'ai jamais croisé en tout cas.

A noter que LTRv2 a été utilisé (industriellement) par d'autres organismes en dehors du ministère de la Défense, notamment la DGAC pour le système CAUTRA⁶ et le GANIL⁷ (dans sa première version, mise en service en 1983) à Caen."

En parallèle vous pourriez peut-être consulter Gérard LE LANN directeur honoraire de recherche à l'INRIA. J'ai travaillé étroitement avec lui dans les années 80 mais ce n'est pas pour cela qu'il peut vous intéresser. Gérard a en effet effectué son service militaire au CPM dans les années 70 et il a sans doute croisé JR ABRIAL. Vous pouvez le contacter de ma part.

En vous remerciant de vous intéresser à un centre unique en son genre et qui a évidemment disparu car trop atypique. "

Il me fallait trouver l'adresse de courriel de Gérard Le Lann. J'ai pensé à Jean Bézivin, ancien collègue de Nantes. Et le 22 mai je recevais la réponse de Gérard Le Lann⁸

5. "Aujourd'hui, l'usine Airbus de Nantes Bouguenais est issue, en 1936, de la volonté de Louis Breguet, l'un des pionniers de la construction aéronautique française, de développer la construction d'avions sur la région nantaise. Un ouvrage intitulé : Tous les Airbus naissent à Nantes rappelle cette saga industrielle dans la rubrique : de Louis Breguet à Airbus." in <https://www.clubfranceinternational.com/archives-et-vestiges-des-usines-daviation-breguet/>

6. Système de Contrôle Automatisé du Trafic Aérien

7. Grand Accélérateur National d'Ions Lourds

8. Pionnier d'Internet, Gérard Le Lann a commencé sa carrière au CERN avant d'intégrer le projet Cyclades à l'IRIA (aujourd'hui Inria) ; il rejoint Stanford par la suite où il travaille avec Vinton Cerf sur Arpanet. Après ses aventures outre-Atlantique, le spécialiste des systèmes distribués a poursuivi son travail chez Inria où il est aujourd'hui directeur de recherche émérite, in Gérard Le Lann : de l'invention d'Internet à l'intelligence algorithmique, INRIA, 11 juin 2024

5.2 Témoignage de Gérard Le Lann

Merci à Jean (Bézivin), bonne et ancienne connaissance du temps où nous étions tous deux à l'IRISA (1972-1978 pour ce qui me concerne). Et merci à Pascal (Grojean), qui fut un exceptionnel partenaire dans l'aventure "Ethernet déterministe" conduite par Inria, en coopération très étroite avec la Marine Nationale (voir pj).

C'est au CPM (rue Sextius-Michel, Paris 15eme) que j'ai fait la connaissance de Jean-Raymond vers 1968, où j'ai effectué une grande partie de mon Service Militaire en tant qu'appelé scientifique (Brest St Renan était mon autre "base"), et membre de l'équipe chargée de concevoir et développer les logiciels destinés à l'informatique embarquée dans le 1er sous-marin SNLE (Le Redoutable). Un quadri-processeur à base de TRW 133 (les mêmes que ceux utilisés par l'US Navy), dont il a fallu modifier l'OS pour remplacer l'algo d'ordonnancement d'origine (périodique) par un algo plus performant (mélange de périodique et de EDF⁹). A cette époque, LTR n'était pas encore "prêt". Jovial¹⁰, etc., langages abandonnés pour finir avec un langage assembleur et une programmation en logandes/logigrammes¹¹.

Magie des rencontres, même brèves ? Quelques discussions au hasard, dans les couloirs ? Jean-Raymond et moi-même sommes restés régulièrement en contact depuis. Il fut l'un des tout premiers à comprendre une distinction fondamentale en "sûreté informatique" : la plupart des échecs ne sont pas causés par des erreurs en "génie logiciel", mais par des erreurs en "génie système" / si les spécifications de départ (capture des exigences, etc.) son erronées relativement aux véritables besoins, on implémente, en logiciel, des solutions non satisfaisantes, même en recourant à des méthodes "formelles". Pas étonnant donc que la Méthode B ait été bien supérieure aux méthodes de type "model checking". Jean-Raymond m'a très fortement soutenu lorsque j'ai publié mon analyse des causes de l'échec du tir de qualification d'Ariane 5 en 1996 (voir pj), basée sur cette distinction.

Félicitations d'Alain Bensoussan, qui venait de quitter sa position de Président de l'Inria pour devenir ... Président du CNES, maître d'œuvre avec l'ESA du programme Ariane 5. C'est l'une de mes deux publications les plus citées, encore actuellement. www.inria.fr.

9. Earliest Deadline First scheduling (« échéance proche : préparation en premier » en anglais) est un algorithme d'ordonnancement préemptif, à priorité dynamique, utilisé dans les systèmes temps réel. Il attribue une priorité à chaque requête en fonction de l'échéance de cette dernière, les tâches dont l'échéance est proche recevant la priorité la plus élevée.

10. Langage dont le premier compilateur a été produit par la société ACT à NYC

11. Le Service Technique des Constructions et Armes Navales (STCAN) est un établissement de la Direction des Constructions Navales ; il est chargé des études et de la conception des navires de la Marine Nationale. Au STCAN comme jeune ingénieur du Génie Maritime, je remplissais la fonction d'ingénieur d'études avec responsabilités techniques comprenant l'étude et la réalisation du cahier des charges, la spécification et l'acquisition de matériel, l'analyse fonctionnelle, la programmation et la mise au point de programmes (programmation de calculateur TRW133 et CAE133, en logandes, "micropogrammes au niveau des instructions de la machine", et logrammes, sous-programmes composés de logandes et chargés de réaliser les instructions habituellement cablées"). J'étais rattaché à Henri Boucher, à l'époque ingénieur en chef du Génie Maritime. Années 63-68, Claude Kaiser, <https://cedric.cnam.fr/~claude/CKCVDetails2009.pdf>

<fr/fr/gerard-le-lann-ethernet-informatique-reseaux>
The Ariane 5 Flight 501 Failure, A Case Study in System Engineering for Computing Systems, Gérard Le Lann, Rapport de recherche INRIA, N°3079, Décembre 1996

Toutes ces informations ont été reçues en moins d'une semaine! Nos vifs remerciements à mes correspondants.