

ROOT, logiciels d'analyse du CERN : comment finir par dire non à l'Europe.

Mon Psy m'avait pourtant dit d'arrêter avec ROOT... Bon enfin, allons-y.

J'ai beaucoup apprécié les réactions de ce forum car cela rejoint un combat que je mène avec d'autres (trop peu) depuis maintenant plus de 20 ans pour le cas PAW et 10 ans pour le cas ROOT. En gros, comment faire pour que le CERN, un laboratoire qui prétend avoir le « E » d'Européen dans son nom et qui va bientôt être, avec le LHC, au coeur de la physique mondiale, arrête de produire du logiciel pour la physique qui est une honte pour l'humanité ?

D'abord il faut démolir les idées reçues ; ROOT n'a pas été « pensé » par des physiciens. Le drame est qu'il a été pensé par de mauvais informaticiens qui ont la chance d'être « sur place », donc proche des physiciens et là où sont produites les données. Ces gens, donc dans une position de facto dominante, sont hélas de très mauvais ingénieurs et de plus en position idéale pour faire passer à peu près n'importe quoi ; c'est le « syndrome Bill Gates » bien connu des gens qui se battent pour avoir enfin un operating system (oups, un « système opératif ») supportable. Ce syndrome est en gros celui de quelqu'un qui arrive avec un bout de code mal fichu mais qui a le bon goût d'arriver au moment opportun pour capturer une niche écologique importante. Pour ROOT, le parallèle avec Windows est vraiment saisissant. Voir ma page « HEP soft-war ? » à (pub) <http://OpenScientist.lal.in2p3.fr> pour une comparaison.

Donc ROOT n'a pas été pensé par des physiciens mais bien par de mauvais « ingénieur logiciel » (je préfère ce terme à « informaticiens ». Sans être péjoratif, ce n'est pas le même boulot d'écrire du logiciel scientifique que de faire de l'exploitation ; voir ma diatribe sur la question à la section « HEP soft-war ? Software for Physics Department in labs »).

Du fait de l'emplacement stratégique du garage où il a été mal conçu (donc près du CERN), ROOT a profité de contribution multiple de certains physiciens et de 10 ans d'input des physiciens sur PAW. Donc au premier ordre, il est vrai que ROOT fait 99% du boulot que demande un physicien. Et alors ? Et alors il est vrai aussi que Windows finit par faire 99% du boulot que l'on demande à un OS, mais que tout le monde est d'accord pour dire que l'humanité se portera mieux lorsqu'on en sera enfin débarrassé. Un machin mal fichu demeure un machin mal fichu et un jour ou l'autre cela vous saute à la figure. Donc il faut s'y coller et le plus tôt sera le mieux.

À propos des physiciens, il est à noter que j'ai dit « certains ». Étant dans un laboratoire HEP non-CERN, je côtoie tous les jours des gens qui s'arrachent les cheveux avec ROOT et souvent les laptops (oups, les « sur les genoux ») volent contre les murs. D'ailleurs, depuis 1995 où ROOT a commencé à polluer nos machines, je suis à peu près sûr que le taux de commandes de « sur les genoux », c'est le cas de le dire, a progressé, chez nous, de manière significative.

Mais à propos de mes chers physiciens, il y a tout de même quelque chose que je ne comprends pas. En gros, en HEP nous récupérons pas mal « de crème » : X, normalien, centralien, etc... En principe, on s'attendrait à ce que ces gens aient deux brins de jugeote sur les questions techniques. Mais de manière surprenante, quasiment aucun ne se lève en place publique pour faire quelque chose pour avoir du meilleur logiciel ; pour ROOT, tout le monde supporte sans broncher. Curieux, ceci est probablement à corrélérer avec le passage très tardif chez nous du FORTRAN à des langages orientés objets. HEP a fini par percuter dix ans après le reste de l'humanité, bizarre. Des fois il y a des choses qui m'échappent. Soudainement, quand on parle logiciel les gens deviennent incroyablement bouchés (restons

polis) et la « crème se met à tourner ». Si un sociologue ou un psychologue cherche un sujet de thèse, il y a matière en HEP... (au fait HEP = High Energy Physics).

Pour les gens qui ne sont pas convaincus que ROOT est mal fichu, je peux en apporter une preuve simple ; l'existence de la méthode `TClass::Draw()` qui résume à peu près tout. En gros ces mauvais ingénieurs ne sont pas fichus de faire la différence entre un système d'introspection et un package graphique ! Pour que tout le monde suive ; en gros un système d'introspection est un truc basique important, qui hélas manque cruellement à C++ (sans commentaire). Cela permet, entre autres, par programme, d'interroger une classe sur son héritage, ses gentils membres, ses méthodes (pas catholiques pour ROOT), etc... Ceci permet de faire beaucoup de choses intéressantes comme, par exemple, d'écrire des adaptateurs de manière génériques pour faire le lien des classes « data » (description des données dans un détecteur par exemple) avec des « facilités » comme, par exemple, un système de stockage, où de scripting ; par exemple enrober (?) du C++ vers du Python. (Voir ma page architecture sur le modèle BAF ou SAF). Fort bien, donc l'introspection est basique et devrait venir avec tout langage OO (donc gros raté du côté C++). Ici soyons honnêtes, ceci a été bien vu du côté du garage qui a vu l'intérêt de coupler l'introspection avec le stockage pour produire automatiquement des streamers pour des classes data (ou user) ; d'où CINT et le TClass, très bien.

Mais peut-on m'expliquer pourquoi avoir une méthode `Draw()` dans une classe `Class` ???

Le fond du problème est que malgré tout, ces garagistes ne sont pas capables de séparer les problèmes et de comprendre qu'il faut du logiciel pour gérer le stockage des données et d'autres logiciels pour gérer le graphique ou l'interface utilisateur. C'est la base du processus scientifique que de séparer les problèmes. Un symptôme de l'esprit intriqué, au sens quantique, de ces gens est par exemple le lien avec CINT.

CINT (pour Crashy INTerpreter ?) est un interpréteur C++ qui a été développé hors ROOT et est hélas distribué maintenant à travers ROOT. (Ne cherchez pas de site pour CINT, il n'y en a pas, et c'est dommage). CINT en lui-même pourrait être intéressant pour plein de chose hors ROOT, et donc il semble raisonnable de dire : je prends la distribution standard de CINT, je l'installe hors ROOT, et après je fais une installation de ROOT en s'appuyant sur l'installation standard de CINT. Ce que tout le monde fait avec zip, freetype2, OpenGL, X11, etc... Mais non ! Avec CINT, ça ne marche pas ; les garagistes ont réussi à polluer CINT avec du code spécifique pour ROOT, et donc CINT doit être construit de manière spéciale pour ROOT (il y a des cpp : `#ifdef G_ROOT` dans CINT !). Donc CINT est maintenant intriqué, pollué par ROOT et il faut faire une installation spéciale pour CINT. C'est la même chose que si en gros le « ROOT core team » s'était acoquiné avec Brian Paul, développeur de Mesa (free OpenGL) ou Guido Van Rosum, développeur de Python, et avait fini par les convaincre qu'il faut des « features » spéciales pour ROOT de tel manière que maintenant il faudrait une installation spéciale d'OpenGL ou Python pour que ROOT marche bien. Tout le monde trouverait cela aberrant ; mais non, pas au CERN ! Et personne ne bronche. Et le cas CINT n'est pas isolé. Le problème c'est que tout est comme cela ; bref, au secours !

Un truc conceptuellement super dans ROOT est la « g intrication ». En gros les « services » du « framework » sont accessibles par des pointeurs globaux ; les `gXxx` (`gSystem`, `gStyle`, `gDirectory`, etc...). Et chaque T-classe d'un « g service » utilise les autres « g services » à travers cela. En gros l'encapsulation, une base de l'orienté objet, est cassé à la base. Pour le développeur c'est un cauchemar à suivre. En particulier il y a des relations entre classes qui n'apparaissent jamais à travers les méthodes et donc impossibles à suivre, par exemple, avec Doxygen. Et je ne

parle pas des liens fait avec des strings (non pas ceux la) en passant par l'interpréteur (donc par gInterpreteur !). (Des strings qui ne sont même pas des std::string (il y a des C string, donc strcmp, etc... !) et, forcément, des TStrings). Si quelqu'un vous montre un diagramme des classes de ROOT fait avec Doxygen, ATTENTION ; une bonne partie de l'iceberg est invisible !

L'équation $ROOT = T-tanic$ (ou $T-panic$ d'ailleurs) n'est, hélas, pas une blague. Tiens, à propos de blagues sur ROOT, si vous en avez, je les collectionne (voir ma section « Anti-Dépression » sous « HEP soft-war ? »).

Pour l'utilisateur c'est aussi très difficile à suivre ; on ne sait jamais trop où on en est puisque un objet où un « g service », par un changement instantané de son état, a potentiellement un impact sur tout les objets ! Un symptôme clair de cela est que beaucoup de gens préfèrent ne pas exécuter deux fois le même script CINT (.C) dans la même session ; pas mal de gens préfèrent relancer ROOT sur le script. Et le système de commande étant du C++ (donc plein de pointeurs en direct) cela ne simplifie pas les choses ; pas facile de savoir où en est l'interpréteur. Donc un scénario « de la vie de tous les jours » qui consiste à éditer son script avec un éditeur externe puis exécuter plusieurs fois sous la même session du programme, n'est en général pas fiable. Les gens modifient le script et relancent ROOT ! Au moins dans PAW on pouvait faire cela. Et c'est pourquoi les gens préfèrent maintenant un prompt Python.

On pourrait croire que l'on pourrait régler tous ces problèmes en sautant dans un TGV pour Genève pour discuter de manière raisonnée avec ces gens qui sont sensés être représentatifs du meilleur que l'ingénierie logiciel scientifique Européenne peut produire. Mais non, on se heurte à un mur. Et là on tombe sur un problème de fond qui est en fait très similaire avec le cas Windows. Qu'un logiciel (ou autre chose de fabriqué d'ailleurs) ne soit pas parfait, soit. Ici on pourrait paraphraser le gars de Nazareth qui semble avoir été en contact avec un très grand développeur : « que celui qui n'a jamais fait de bug jette le premier sur-les-genoux ». Le problème n'est donc pas le fait qu'il y a des choses à revoir ; le problème est que le processus scientifique de critiquer par les paires ou collègues ne fonctionne pas avec ROOT. Jamais une « software review » n'a été faite par des paires ! Les garagistes se sont toujours arrangés pour que les pseudos revues soient en fait une sorte d'opinion d'un groupe d'utilisateurs physiciens évidemment très choisi.

Je proposerais même un truc. Je propose, donc publiquement, qu'une revue de ROOT soit faite avec dans le comité : Richard Stallman, Bjarne Stroustrup, Guido van Rosum, un gars de SUN ayant participé au design de java, Josie Wernecke (Open Inventor Architecture Group SGI), le gars qui a fait MySQL. Je ne veux personne ayant participé à l'écriture de Windows puisque personne n'a jamais pu voir les sources de Windows. Si ces gens arrivent à la conclusion que ROOT est une percée majeure de l'humanité pour le logiciel, alors d'accord, je m'écaserai jusqu'à ma retraite.

Évidemment, les physiciens sont écoutés à propos de nouvelles fonctionnalités, mais pas les gens qui sont payés pour avoir un regard sur le « comment c'est fait » et qu'on appelle en bout de chaîne lorsque l'utilisateur ne s'en sort pas. En gros, tout ingénieur logiciel arrivant avec des critiques sur le « comment c'est fait » est vu, détecté comme un « ennemi a la cause » qui doit en gros être éliminé. Cela ne vous rappelle pas quelque chose ? Et là, je pense qu'un psy aurait pas mal à dire.

Ce comportement profondément fraternel, qui est un des fondement économique des ex colonies anglaises, est peut-être la règle du jeux dans les compagnies privées ou dans des laboratoires à tendance très nationale mais, est évidemment totalement inacceptable venant d'un laboratoire (donc le CERN pour les retardataires) qui a été créé pour fédérer les ressources, et donc l'ingénierie, d'une collection de pays Européen pour

faire de la science ; ressources que chacun ne pourrait pas se payer seul. Quelque part un comportement « à la Bill Gates » vis à vis des instituts des pays membres du CERN est en gros « anti CERN constitutionnel ». Intéressant n'est-ce pas ? Je suis sûr que certains d'entre vous ont en tête cette image d'Epinal du CERN comme une grande fraternité de peuples se groupant autour de l'anneau de collision pour faire de la science et où, en gros, il ne manque plus que le feu de camp au milieu et Hugues Aufray à la guitare (et Hisse et ho).

Quand on « non discute » avec le chef garagiste, il est surprenant de voir qu'il a en tête le modèle Boeing alors que, vu qui le paye, il devrait avoir en tête le modèle Airbus ! (Et qui d'Airbus ou Boeing va maintenant avoir la plus grosse... capacité de transport ?) (À propos de salaire, je n'ose même pas comparer les salaires CERNois avec le mien. D'ailleurs je ne vais plus au CERN avec mon Scenic ; même propre (il faut bien, c'est la Suisse), cela fait tache sur le parking).

Il est clair que maintenant la situation est bien bloquée. Comme Bill, les garagistes se protègent maintenant derrière le nombre de leurs pauvres utilisateurs physiciens. Et comme étrangement ceux-ci s'écrasent (problème de gros sous ?) (ou bien on ne touche pas au CERN ?) et bien rien ne se passe.

En tout cas, si certains de mes concitoyens ont voté en Mai pour moins d'Europe dans leur vie sans trop savoir pourquoi ; je peux maintenant leur donner de bonnes raisons.

En fait, après une longue réflexion sur le sujet, je pense maintenant comprendre ce qui ne va pas, et qu'en fait le problème est plus profond que l'on ne pense. En fait l'activité « développeur, ingénieur de soft pour la physique (ou une science en général) » n'est en gros pas reconnue en tant que tel. Tout est encore du bricolage d'individu et en gros les gens (crème ou pas) finissent par être content de trouver quelque chose qui fonctionne, même à peu près. Le code est soit produit par des physiciens qui devraient normalement concevoir des détecteurs ou buller sur la nature du monde ; ou bien produit au sein de « service informatique » par des ingénieurs n'ayant souvent pas de vue sur la physique. Et il est clair que le courant a du mal à passer. En gros on voit souvent débarquer, donc au service « informatique », des physiciens disant « mon programme Geant4 ne reproduit pas correctement les courbes de physique hadronique de diffusion de π sur des protons » ; peux-tu m'aider à décoincer cela ? Bien sûr très cher ; heu, as-tu vérifié que ton π vaut bien 3.14 ? Et croyez-moi que les gens qui sont capables de faire face ne courent pas les rues. Dans mon service (donc informatique) on nous appelle gentiment les « super développeurs ». (D'ailleurs si un de mes super chefs lit cela, il serait sympa de penser à mon super salaire, au moins pour avoir la super bagnole qu'il faut pour aller en mission au CERN).

En gros, le code n'est toujours pas produit par des personnes qui seraient donc des professionnels du soft mais ayant aussi un regard sur la physique. En fait la notion de « service développement logiciel pour la physique » est un chaînon manquant dans toute cette histoire et un jour, il faudra bien faire quelque chose (Vois ma section « Software for Physics Department in labs »). Ce manque existe aussi au CERN. En fait, là-bas, on ne sait pas où caser les « développeurs pour la physique » ; chez CERN/IT ou bien dans les expériences ? Il est intéressant de noter que le chef garagiste a souvent fait la navette IT / expérience (et donc a fini par déprimer dans... son garage !). (La dépression, c'est terrible, c'est incroyable les bêtises que l'on peut faire dans ces cas là).

En fait CERN/IT (300 personnes !) n'a clairement plus voix au chapitre et est clairement dédié exploitation (donc du GRID maintenant). Au point de vue logiciel pour la physique, les pauvres sont coincés à maintenir CERN/PAW (parce qu'il reste pas mal d'utilisateurs dans le monde), alors que

les développements de ROOT se font en fait spirituellement dans un garage. (Donc lorsque l'on dit que ROOT est maintenu, développer par le CERN ; il faut faire attention à ce que l'on dit et surtout à ce que l'on achète ; rien n'est clair sur ce point, surtout si CERN/IT est censé représenter la voix du soft !).

Quant aux expériences (donc pour le LHC ; ALICE, ATLAS, LHCb, CMS), elles sont totalement incapables de mettre une ligne de code en commun. (Et c'est dommage parce que ça draine pas mal de crème). Pour les expériences LHC, il a été finalement fait un truc appelé « LCG » pour tenter de mettre du soft en commun ; pour l'instant le résultat est en gros une concaténation des bases des softs des quatre expériences. En gros la synergie conceptuelle plus la mise au propre, qui aurait dû être faite pour partager les efforts, n'a pas été faite ; et le CERN porte une grosse part de responsabilité la dedans (rappelez vous de ce que veut dire « E » dans CERN). Et soyez sûrs que chacun prend bien soin de ne pas dépendre du soft des autres. En langage physicien, on peut résumer cela en disant que LCG est une collaboration « fermionique » alors que cela aurait dû être, depuis au moins dix ans, une collaboration « bosonique ». (En fait, qu'en on y pense, l'Europe c'est assez bien cela ; une fédération fermionique d'états).

Qui est responsable ? Les garagistes y sont pour beaucoup, ça c'est sûr. Mais en fait, tout le monde et personne ! En fait le principal problème est que, historiquement, on n'a pas compris que le « logiciel pour la physique » est une activité en soi à mettre en parallèle avec l'électronique, la mécanique, l'exploitation. Et en particulier que ce n'est pas un sous-produit de l'exploitation comprise comme « informatique » (informe et à tiques). Dans mes pages, je fais l'analogie avec Harry Potter ; écrire l'histoire d'Harry Potter n'est pas le même boulot que d'imprimer les millions de livres pour la planète. Historiquement, lorsque les laboratoires de physiques (par exemple à Orsay) ou bien le CERN ont été créés, les gens ont mis les « écrivains » dans le même giron que les « éditeurs ». (En fait, ils avaient même pas conscience qu'il y avait des « écrivains »). Et PAW, ROOT viennent de cela.

Qui aurait pu forcer une refonte à temps de ROOT ? Seule une entité, comme une « Software for Physics Division » composée d'écrivains professionnels et avec suffisamment de bouteille pour être écouté par les expériences, et surtout avec sa voix propre dans les divers conseils du CERN aurait pu faire cela. Manque de chance, cette entité n'existe pas ! (Et après tout, qui peut blâmer les garagistes de profiter de la situation ? Comme pour Windows, tant qu'il y a des gens à la vue basse pour acheter...)

LCG est peut-être un début de cette entité, mais cela arrive trop tard pour le LHC, et les garagistes ont tout fait pour cela. En particulier le CERN va rater le premier système de stockage de masse OO open source bien fichu. Tant pis, mais bon, en fait ce labo est déjà connu pour être passé à côté de pas mal de choses et j'ai l'intime conviction, qu'il aurait pu inventer l'OO et un langage comme java (il y a eu une période où des gens écrivaient des compilateurs au CERN). Bon il y a eu le web, mais j'aimerais bien avoir l'opinion profonde de Tim Berners-Lee sur le CERN. (Tiens, d'ailleurs pourquoi il n'est plus au CERN celui-là ?)

Donc le soft du LHC va être une honte technologique. Mais bon, ne dramatisons pas, si tout le hardware est bien fait, et le Higgs est au rendez vous, même un soft bien pourri devrait pouvoir tomber dessus. Mais il serait tout de même quelque part immoral que ce soit un comportement « à la Bill Gates » et surtout le soft qui va avec qui plot le Higgs.... En tout cas lorsque vous verrez des plots de Higgs partout dans les revues et que vous reconnaîtrez que c'est fait en ROOT (c'est facile à repérer) au moins vous, vous saurez de quoi il en retourne. (En fait le CERN devrait

imposer que les plots pour les revues soient au moins faits en GNUplot, ou n'importe quel logiciel de plotting dédié).

Maintenant que nous sommes fixés sur le LHC, il faut voir ce que l'on peut faire pour l'après LHC (Linear Collider, et tous les programmes qui ne sont pas rattachés au CERN). Pour ceux que cela intéresse de voir ce que je propose, voir OpenScientist donc à <http://OpenScientist.lal.in2p3.fr> . (Ça mûri, ça mûri. Ça va bientôt être très buvable).

Tiens, faites moi penser un jour de parler des softs de simulation en HEP (Geant4, FLUKA, VMC, etc...). Là aussi il y a des pauvres utilisateurs qui s'arrachent les cheveux. Il y a pas mal à dire (et encore une fois on tombe sur les mêmes pollueurs).

Système de stockage OO ; appel d'offres :

Oui, donc le CERN va nous rater un système de stockage de masse OO open source bien fichu et donc le problème reste entier (sauf que maintenant j'ai Rio, mais cela ne suffit pas). En gros il me faudrait une couche d'IO genre HDF5 (ou Rio) sur lequel on aurait rajouté une couche OO grâce à un système d'introspection C++ (sans méthode Draw évidemment) avec lequel on pourrait déclarer des classes « user » et qui nous produirait les streamers pour la couche de base (par exemple HDF5 ?). En plus il faudrait que le streaming soit compatible avec un parallèle fait en java (parce que mes copains du groupe AIDA de SLAC sont en java). Si les streamers pouvaient être « pure abstract », cela serait génial. Et le tout en open source, tournant donc au moins sur les trois « sur les genoux » du moment (Linux, MacOSX, et le Win machin de l'autre). (Du transactionnel ne m'intéresse pas et a priori du relationnel non plus).

Si quelqu'un a quelque chose... Sinon tant pis, comme d'habitude je finirai bien par faire le boulot. La retraite est encore loin.

Voilà, voilà. Bon il faut que je retrouve mes anti-dépresseurs et que je m'y recolle. (En plus, avec tout ça, je viens de rater poubelle la vie sur la 3).

(Tiens, en fait ça donne des idées. On pourrait proposer aux scénaristes des épisodes où l'affreux Frémont commence à installer ROOT sur toutes les machines du quartier du panier : enfin un vrai virus. À creuser...)

Remerciements à mon jeune Padawan pour être tombé sur ce forum. Le peu de fois où il va seul au CERN, je m'inquiète qu'il ne croise à la cantine l'empereur Sith ou Vadehors ; il n'est pas encore prêt... (qui l'est d'ailleurs ?)

Pour les gens non HEP : la cantine du CERN est évidemment un endroit hautement stratégique sur cette planète ; c'est là qu'est fait le vrai travail de pub, de racolage et surtout de démolition de tout produit concurrent et des bonshommes qui vont avec ; là, on voit le vrai visage de l'Europe.

URL : <http://OpenScientist.lal.in2p3.fr>