

## Passage aux techniques OO dans H1

- Echelle de temps
- Motivations, contraintes
- L'analyse dans H1
- Approches choisies (3 phases)
- Applications existantes (phase 1)
- Tag data base (phase 2)
- Projet OOTWG (phase 3)
- Conclusions

## Echelle de temps:

- été 2000 : upgrade du collisionneur en luminosité:
  - changements importants dans le détecteur
  - le volume de données par an ne va pas augmenter (nouveaux mécanismes de trigger), mais le volume global va devenir vraiment important
- 2005: fin de prise de données

D'où:

1. le soft H1 doit tenir encore au moins 5 ans
2. il doit pouvoir être adapté aux nouveaux détecteurs
3. il doit tenir le choc avec un volume de données de plus en plus important

Et:

**Puisque de toute façon on sera obligé de changer des choses, n'est-ce pas le moment pour prendre le train et profiter des avantages des nouvelles technologies?**

## Motivations, contraintes

### Le soft peut-il tenir jusqu'en 2005??

- les packages utilisés (PAW, Geant, Zebra, Cernlib) ne seront plus maintenus
- la maintenance du soft H1 devient difficile ( trop d'historique, gens partis)
- l'adaptation a HERA 2000 ne sera pas évident
- les nouveaux outils très attractifs sont difficilement utilisables dans notre cadre actuel
- une expérience qui ne fait que du Fortran devient inattractive pour thésards, informaticiens
- déficiences de l'analyse actuelle

### Mais:

- on ne peut certainement pas réécrire tout le soft
- la compatibilité arrière est essentielle (logiciel, accès aux données )
- les gens ne sont pas formés (et pas nécessairement volontaires)

## L'analyse dans H1 actuellement

### Faits de base..

- Volumes: POT 7 TB/an, DST 500 GB/an
- accès aux données via Fpack (Fortran, fait maison, accès essentiellement séquentiel mais indexfiles existent)
- lecture d'événements complets uniquement
- les DST ne contiennent pas toutes les relations intéressantes// (hits ↔ tracks, etc.)
- le re-calcul d'une variable physique entraîne la réécriture de l'événement complet

**Exemple: la régénération des DST et des event-directories prend plusieurs semaines**

## Approche choisie

- l'endroit essentiel ou il faut faire quelque chose, c'est l'analyse
- par chance, c'est aussi le sujet ou il y a les nouveautés les plus tentantes
- le plus important est de changer l'accès aux données

## Approche en 3 phases:

1. actuellement: petits essais individuels, familiarisation avec les outils
2. futur proche: tag data base
3. futur plus lointain: étude d'un modèle de données complet

## phase 1 : choses existantes

- monitoring online de composantes hardware du détecteur (C++)
- monitoring online de la qualité de certains triggers et classes d' données (JAVA)
- prototype d'un offline event display en RooT (intégration de l'event display dans un cadre général d'analyse, gain en interactivité)

## phase 2 : tag data base

Largement inspiré de l'expérience positive de ZEUS.  
Actuellement ell est en préparation.

Le but est:

1. sélection rapide dévénements
2. analyse rapide de qualité des événements
  - Choix techniques: Objectivity  
interfaces : Fpack, RooT, JAS, web-tools
  - Interface fpack via expressions de sélection: OO  
complètement caché
  - Variables: identification des événements ,variables  
cinématiques et topologiques, final state objects, variables  
de contrôle En tout  $\sim 100$  variables/mots (Zeus en met  
 $\sim 200$ )

On gagne:

- l'accès rapide aux quantités sélectionnées pour le contrôle  
de qualité
- sélection rapide d'événements sur des critères complexes
- flexibilité de stockage pour des quantités qui risquent d'être  
recalculés(tracks, jets, particules...)

**Zeus estime avoir gagné un facteur 5 à 10 d'accès aux  
variables sélectionnées**

## phase 3: étude d'un modèle de données

### OOTWG: =Object Oriented Technology Working Group

- collaboration entre H1 Zeus et ZDV, d'autres expériences de DESY pourront s'associer
- doit commencer le travail cet été, pour 1,5 années
- 1 postdoc H1, 1 postdoc Zeus, 1 postdoc ZDV + project leader, + personne de conseil extérieur d'une boîte commerciale

### BUT:

- développer un modèle de données aussi commun que possible pour les deux manip qui servira pour stocker les événements dans une database OO
- utiliser
  1. technologie OO pour la database d'événements (très probablement Objectivity)
  2. des outils de design OO (lesquels?)
  3. du project management

**C'est un essai: on admet qu'on jettera tout si ça ne donne pas satisfaction**

## Conclusions

- Il n'est pas évident, ni au niveau technique ni au niveau sociologique, de transformer des structures existantes depuis plus de 10 ans
- Malgré cela, l'équipe dirigeante de H1 a une ferme volonté d'essayer une transition vers les techniques OO pour H1 après l'upgrade de l'an 2000
- Il semble qu'avec l'arrivée du nouveau spokesman cet été, une décision de principe puisse être prise dès septembre
- On gardera nécessairement certaines partie du soft en Fortran (reconstruction,...). Les changements concerneront essentiellement l'accès aux données et l'analyse
- Pour les gens récalcitrants il y aura une possibilité d'accès aux données sans connaître du OO
- On essaie dans la mesure du possible d'associer Zeus et les autres expériences de DESY à l'effort de migration