

Structures de données et compilation des LSPL

d'Ariane-Y

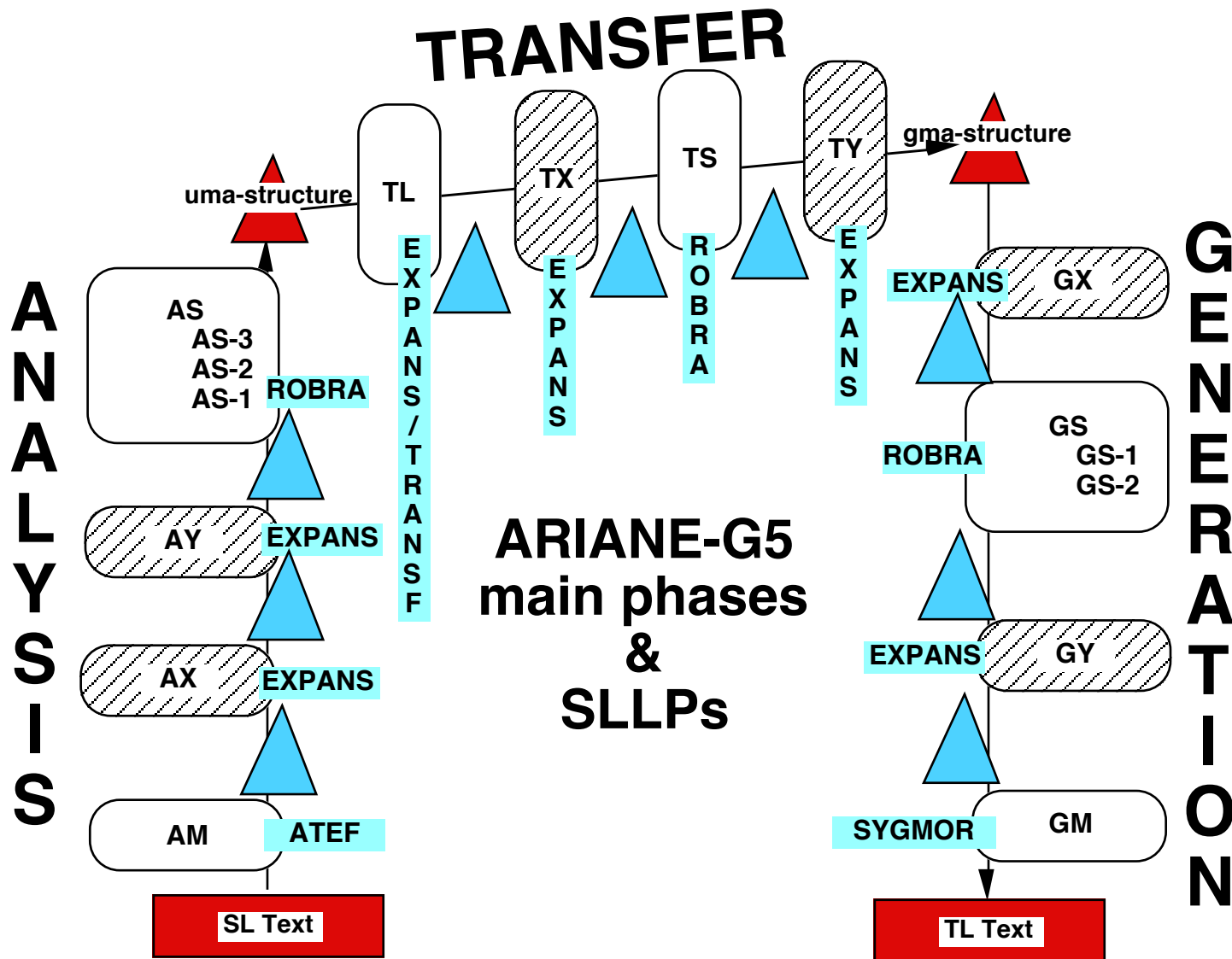
Jean-Claude Durand

- Ariane-G5
- Ariane-Y
- Bibliothèques-AY
 - TableauFichier
 - Liste, Pile, Arbre
- Compilation
 - ANTLR
 - Client-serveur

Ariane-G5

Environnement de développement et d'exécution de systèmes de TA

- 5 langages de haut niveau : LSPL
 - ATEF analyse morphologique
 - ROBRA analyse syntaxique et transfert
 - TRACOMPL transformations complémentaires
 - EXPANS outils dictionnaires
 - SYGMOR génération morphologique
- un moniteur gestion des textes et des linguiciels
- un environnement réseau (smtp, http...)
- sur IBM sous VM/CMS en ASM, PL360, PL/I, exec2



13 compilateurs pour les ≠ "composants"

| Type de composant | TRACOMPL | EXPANS | ROBRA | ATEF | SYGMOR | |
|---------------------------|----------|-----------------|-------|---------------|--------|-----|
| variables | DV | DV | DV | DVM, DVS | DV | |
| formats simples | | FTCD, FTAF | FTAF | FTM, FTS | FTAF | |
| formats composés | | | | FTG | | |
| procédures de décorations | | PCP, PCIS, PRCA | | | PROC | |
| grammaires | | | GR | GR | GR | |
| dictionnaires | | DICG | | DEA, DEB, DET | DEL | DEV |

Effort cumulé de développement

4 étapes principales depuis 1971: Ariane-78,
Ariane-85, Ariane-G5, Ariane/LIDIA

- Compilateurs 20 hxa
- Moteurs 5 hxa
- Moniteur 15 hxa
- Réseau 5 hxa

Arbre décoré

```
1:'ULTXT'(2:'*GN'(3: '.',
           4: '*ANLAGEKOSTEN',
           5: '.'))
1 ' ': UL('ULTXT').
2 '*ANLAGEKOSTEN': UL('*GN'), KSY(GN), P(3), NUM(PLU),
                  KMS(NM), SUBN(PF), PPL(3), TYPO(MJ1).
3 ' ': UL('.').
4 '*ANLAGEKOSTEN': UL('*ANLAGEKOSTEN'), SF(GOV),
                  P(3), NUM(PLU), KMS(NM), SUBN(PF), CPT(NOM, ACC, DAT, GEN),
                  PPL(3), TYPO(MJ1).
5 ' ': UL('.'), KMS(PC).
```

```
<noeud numero="1" FORME=" " UL="ULTXT" >
  <noeud numero="2" FORME="*ANLAGEKOSTEN" UL="*GN"
    KSY="GN" P="3" NUM="PLU" KMS="NM" SUBN="PF" PPL="3" TYPO="MJ1" >
    <noeud numero="3" FORME=" " UL="." />
    <noeud numero="4" FORME="*ANLAGEKOSTEN" UL="*ANLAGEKOSTEN"
      SF="GOV" P="3" NUM="PLU" KMS="NM" SUBN="PF"
      CPT="NOM ACC DAT GEN"
      PPL="3" TYPO="MJ1" />
    <noeud numero="5" FORME="." UL="." KMS="PC" />
  </noeud>
</noeud>
```

Déclaration de variables

** exemple pour PIMLIG, le 20/06/2012 .

-EXC-

\$ENONCP == (** Type d' énoncé .
DECL, INTER, IMPER).

AUX == (ETRE,
AVOIR).

-NEX-

CAS == (** Cas d'un pronom.
NOM, VOC, ACC, DAT, GEN, PREP, RFL).

-ARITH-

NUMERO == (37).

-FIN-

Forme XML

```
<DecVar token="15" Lig="3" Bcol="4" Ecol="11">
  JPGLidia
  <Deco token="19" Lig="6" Bcol="4" Ecol="8">
    lidia
    <Var token="16" Type="TypeExclusif" Etat="Dollar" Lig="9" Bcol="2" Ecol="7">
      ENONCP
      <Val token="24" Lig="10" Bcol="21" Ecol="24">
        DECL
      </Val>
      <Val token="24" Lig="11" Bcol="21" Ecol="25">
        INTER
      </Val>
      <Val token="24" Lig="12" Bcol="21" Ecol="25">
        IMPER
      </Val>
    </Var>
    <Var token="16" Type="TypeExclusif" Etat="Nouvelle" Lig="14" Bcol="1" Ecol="3">
      AUX
      <Val token="24" Lig="14" Bcol="18" Ecol="21">
        ETRE
      ...
```


Règle de réécriture (ROBRA)

RGNGA:

$0(1(2(\$L1)), *, 3(\$L2)) /$

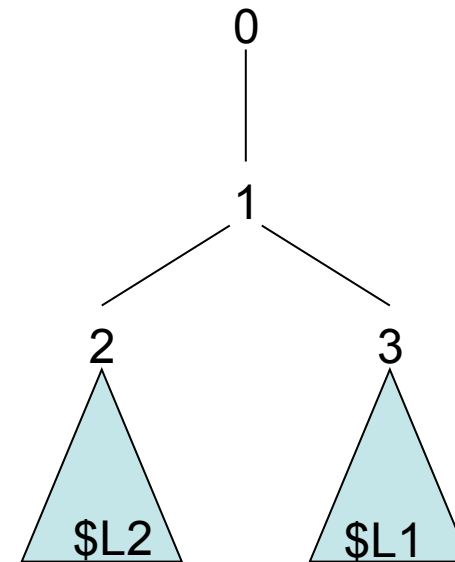
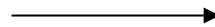
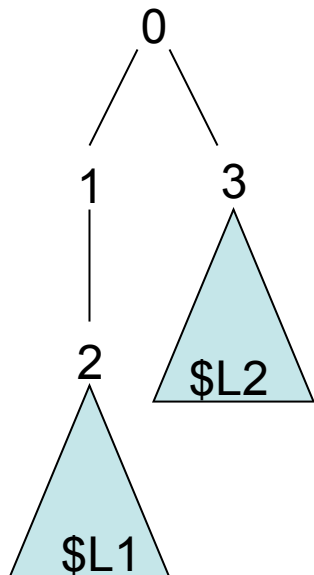
0: UL -E- ULFRA; 1: K -E- GN; 2: FS -E- GOV; 3: K -E- GADJ /

CAS(1) -I- CAS(3) -NE- CAS0

==

$0(1(2(\$L2), 3(\$L1)) / /$

3:3, FS := ATN.



- Grammaire: liste de règles de réécriture
- Graphe de contrôle: grammaires sur les nœuds et conditions sur les arcs

Ariane-Y - Projet Traouiero

- Porter et unifier
 - sur unix/linux
 - en C/C++
- Étendre
 - levée de limites d'implémentation -> gros traitements
 - quelques points de syntaxe
 - cadre moins rigide
 - fonctionnement incrémental + client/serveur
- Contraintes
 - Compatibilité ascendante avec Ariane-G5
 - Équipe réduite

Plate-forme de développement

- Outils make, SVN
- Générateurs d' analyseurs: ANTLR (www.antlr.org)
- Documentation: doxygen (www.doxygen.org)
- Fichiers source et bibliothèques organisés en répertoires

Bibliothèques-AY

Ensemble de classes C++:

- Tableaux-fichiers
 - Tableaux à une dim non bornée
 - À l'aide d'une pagination simplifiée
 - Persistance des données
- Arbres n-aires, binaires, AVL, XML et arbres Ariane
- Ensembles de chaînes
 - factorisation + accès rapide

Compilation

- Générateur d'analyseurs ANTLR
 - Analyseurs LL;
 - en différents langages (Java, C/C++, Python...);
 - Une seule grammaire, ou séparation en deux lexicale et syntaxique;
 - Règle à la BNF (majuscule et minuscule);
 - Actions dans les règles (entre accolades);
 - Prédicats syntaxiques et sémantiques;
 - Langage de macro pour la construction des arbres abstraits et construction automatique; « nœuds imaginaires »;
 - Possibilité de composer les grammaires (>12 pour les LSPL);
 - Interface graphique ANTLRWorks;
 - Communauté des utilisateurs très active.

Compilation (suite)

- Construction incrémentale des structures de données pendant l'analyse avec Bibliothèques-AY;
- Architecture client-serveur;
- Langage de scripts simple obtenu en reportant sur le serveur le contexte d'exécution (espaces de noms, pile des contextes, ...)