

MIGRATION FORTRAN77/ESOPE

SOUTENANCE DE STAGE INGÉNIEUR

YOUNOUSSA SOW

POLYTECH-LILLE

TUTEUR ÉCOLE : FREDERIC HOOGSTOEL

TUTEUR INRIA : NICOLAS ANQUETIL

28 JUIN 2022



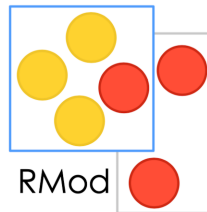
framatome



- 1 Présentation du problème
 - Objectifs
- 2 Identification des analyseurs syntaxiques
 - Définition des critères d'identification
 - Analyseurs Fortran77
- 3 Méta-modélisation
 - Moose
 - Famix
- 4 Démonstration
- 5 Conclusion
- 6 Questions

- Stage ingénieur de 5 mois

framatome



- Détenu par EDF (>75%)
- Acteur international de la filière nucléaire
- Codes de calculs en neutronique
- Equipe de recherche de l'Inria
- Rétroingénierie et évolution des logiciels
- Outils pour le développement
- Langage de programmation Pharo

PRÉSENTATION DU PROBLÈME

PRÉSENTATION DU PROBLÈME

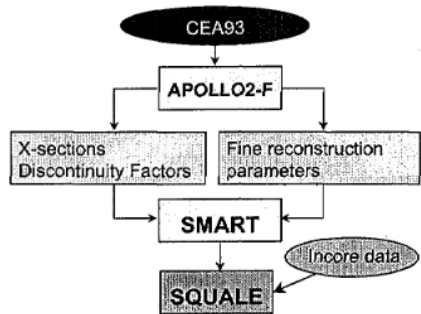
- Fortran77 : langage utilisé pour les calculs scientifiques
- ESOPE : extension du langage fortran 77
 - la notion d'ensemble de données : SEGMENT.
 - les instructions agissant sur le SEGMENT
 - une variable POINTEUR référence le SEGMENT

```
subroutine demo
  integer i
  integer n
  segment foo
    real bar(n)
  endsegment
  pointeur baz.foo
  segini,baz
  i=1
  baz.bar(i)=i*i
  segsup,baz
end
```

Diagram illustrating the Fortran 77 code with ESOPE extensions. The code defines a subroutine 'demo' with integer variables 'i' and 'n'. It declares a 'segment foo' containing a 'real bar(n)' array. A 'pointeur baz.foo' is defined, and 'segini,baz' is used to initialize the segment. The loop 'i=1' and 'baz.bar(i)=i*i' are shown, with 'baz' being a pointer to the 'foo' segment. 'segsup,baz' is used to end the segment. Red boxes and numbers 1, 2, 3, and 4 highlight specific parts of the code: 1 points to the 'segment foo' block, 2 points to 'segini,baz', 3 points to 'pointeur baz.foo', and 4 points to 'i=1'.

PRÉSENTATION DU PROBLÈME : ESOPE

- Utilisé par des codes CEA/Framatome
- 100 ingénieurs travaillent sur ESOPE
- Codes avec une longue durée de vie
- Codes gros
 - SMART : 600 000 lignes
 - 50% en ESOPE
 - 50% en Fortran77 pur



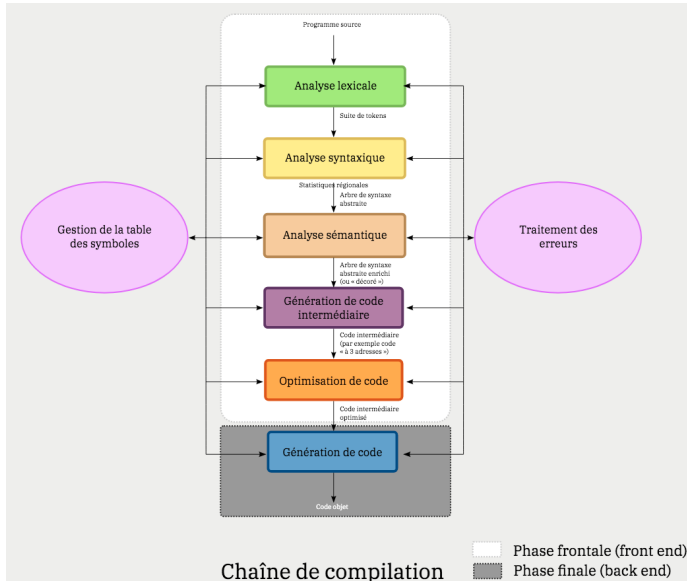
PRÉSENTATION DU PROBLÈME

- Refus d'optimiser le Fortran77 généré
- Le traducteur ESOPE génère des décalages en mémoire
- Menace sur la pérennité des codes
- Traduction manuelle possible mais fastidieuse

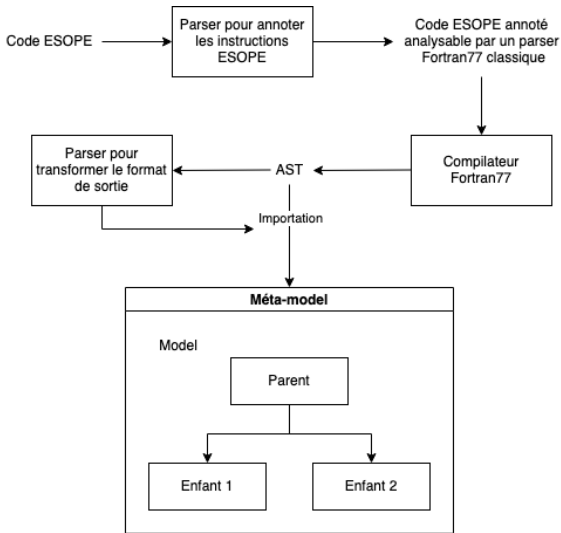
```
SUBROUTINE DEMO
  INTEGER I
  INTEGER N
  COMMON/OOOCOM/OOT,OOV(2),OO_001,OO_002
  ...
  REAL OO_002(2)
    N=3
  CALL OOOWIN(OO4,0,'DEMO 9 BAZ ',OO1,1+N)
  OO_001(-OOO2+OOV(OOT+OO1)+1)=N
  BAZ=OO1
  OO1=BAZ
  CALL OOOWAC(OO4,0,'DEMO 13 BAZ ',OO1,0)
  I=1
  OO_002(-OOO4+OOV(OOT+BAZ)+1+I)=I*I
  OO1=BAZ
  CALL OOOWSU(OO4,0,'DEMO 17 BAZ ',OO1,0)
  BAZ=0
END
```

- Identification d'un analyseur syntaxique de sources Fortran77
- Chargement de la sortie de l'analyseur de Fortran dans Moose
- Construction d'un méta-modèle pour Fortran77

RAPPELS : COMPILATION



SOLUTION MISE EN OEUVRE



EXEMPLES D'ANNOTATIONS ESOPE

| Les types d'instructions ESOPE | Les annotations ou transformations |
|--|------------------------------------|
| - définition de SEGMENT | cESO |
| - manipulation et utilisation des segments (segin, segact, segadj, segdes, segdel) | cESO |
| - déclaration de POINTEUR | cESO |
| - les accès aux dimensions a.b(/1) | ESOsl(a, b, 1) |
| - les instructions d'affectation des champs a.b=var/valeur | ESOat(a, b) = var/val |
| - les instructions de la forme a.b(c.d) | ESOar(ESOat(a,b),ESOat(c,d)) |
| - les notations pointées implicites a = ... | Non pris en charge |

IDENTIFICATION DES ANALYSEURS SYNTAXIQUES

DÉFINITION DES CRITÈRES D'IDENTIFICATION

- Analyse du Fortran77 (**F77**)
- Analyse du Fortran77 généré par ESOPE (**ESP**)
- Donne une représentation intermédiaire facile à manipuler (**AST**)
- Garde les commentaires (**FMT**)
- Garde les positions (**POS**)
- Table des symboles (**TS**)
- A source ouverte (**OS**)

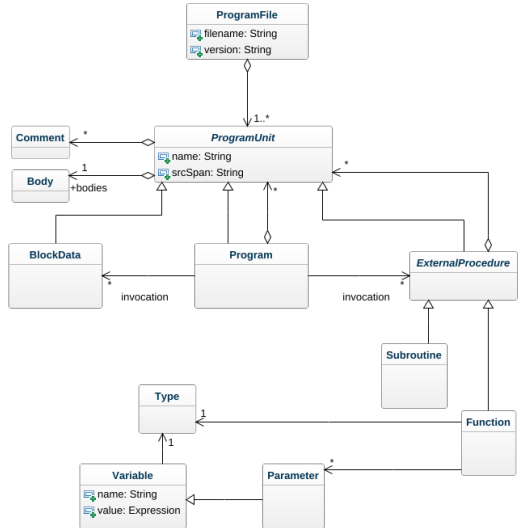
| Tableau récapitulatif de l'analyse des parseurs | | | | | | | | |
|---|--------------------|------|-----|---------|-----|-----|-----|-----|
| Parseurs \ Critères | F77 | ESP | AST | FMT | CMT | POS | TS | OS |
| Gfortran | Oui | Oui | Oui | Inconnu | Non | Non | Oui | Oui |
| Fortran-src | Oui | Oui | Oui | ADT | Oui | Oui | Non | Oui |
| OFPP | Non | Non | Oui | XML | Non | Non | Oui | Oui |
| f2c | Oui | Non | Non | | | | | |
| ANTLR | Oui* ²⁶ | Oui* | Oui | | | | | |
| BNFC | Oui* | Oui* | Oui | XML | Non | Non | Oui | Oui |
| lfortran | Non | Non | Oui | Inconnu | Oui | Non | Non | Oui |
| flang | Non | Non | | | | | | |
| PetitParser2 | Non | Oui | Oui | | Oui | Oui | Non | Oui |

Table 1 – Tableau récapitulatif des différents analyseurs étudiés

MÉTA-MODELISATION

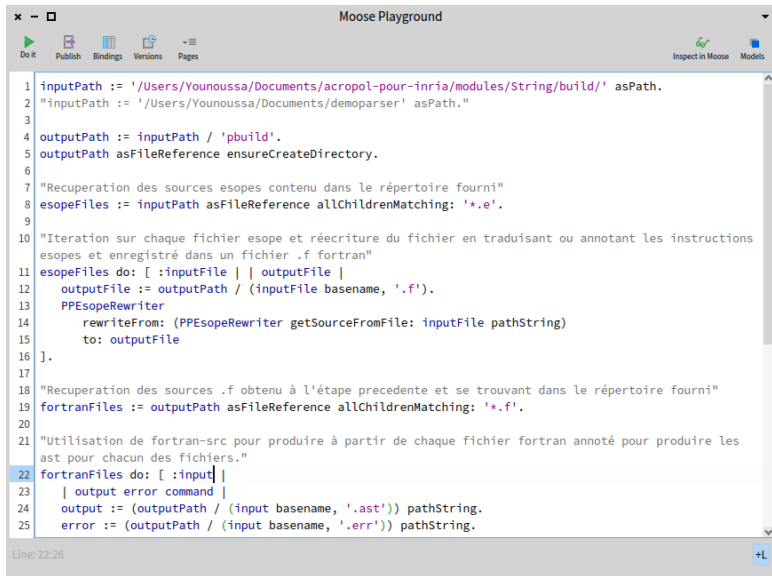
- Une plateforme pour l'analyse des logiciels
- Une plateforme pour le développement d'environnements exploratoires pour tout langage
- Basée sur un méta-modèle générique (Famix)
- Langage DSL pour décrire le méta-modèle

- Utilisation des traits
- Définition des concepts du langage (sous forme de classes)
- Définition de la hiérarchie entre les classes
- Définition des propriétés de chaque classe
- Définition des relations entre les classes



DÉMONSTRATION

DÉMONSTRATION



The screenshot shows the 'Moose Playground' web application. The interface includes a top toolbar with icons for 'Do it', 'Publish', 'Bindings', 'Versions', and 'Pages'. On the right, there are links for 'Inspect in Moose' and 'Models'. The main area is a code editor with line numbers 1 through 26. The code is in English and Fortran, defining paths, file references, and loops for processing Fortran files. The status bar at the bottom left shows 'Line: 22:26' and the bottom right has a '+L' icon.

```
1 inputPath := '/Users/Younoussa/Documents/acropol-pour-inria/modules/String/build/' asPath.
2 "inputPath := '/Users/Younoussa/Documents/demoparser' asPath."
3
4 outputPath := inputPath / 'pbuild'.
5 outputPath asFileReference ensureCreateDirectory.
6
7 "Recuperation des sources esopes contenu dans le répertoire fourni"
8 esopeFiles := inputPath asFileReference allChildrenMatching: '*.*'.
9
10 "Iteration sur chaque fichier esope et réécriture du fichier en traduisant ou annotant les instructions
    esopes et enregistré dans un fichier .f fortran"
11 esopeFiles do: [ :inputFile | | outputFile |
12     outputFile := outputPath / (inputFile basename, '.f').
13     PPEsopeRewriter
14         rewriteFrom: (PPEsopeRewriter getSourceFromFile: inputFile pathString)
15         to: outputFile
16 ].
17
18 "Recuperation des sources .f obtenu à l'étape precedente et se trouvant dans le répertoire fourni"
19 fortranFiles := outputPath asFileReference allChildrenMatching: '*.*'.
20
21 "Utilisation de fortran-src pour produire à partir de chaque fichier fortran annoté pour produire les
    ast pour chacun des fichiers."
22 fortranFiles do: [ :input |
23     | output error command |
24     output := (outputPath / (input basename, '.ast')) pathString.
25     error := (outputPath / (input basename, '.err')) pathString.
```

Line: 22:26

CONCLUSION

- Identification de deux analyseurs
 - gfortran
 - fortran-src
- Création d'un méta-modèle pour Fortran77
- Conception objet avancée
- Interprétation de programmes
- Analyses de programmes divers et variés
- Contact avec des ingénieurs et chercheurs

QUESTIONS