

LOGICIEL : ENSEMBLE PROJET DE PROGRAMME DU MÉTACOMPILATEUR PILOTE-METACOMP

FICHE N° 6966

PRÉSERVER
SAUVEGARDER
VALORISER

Période de fabrication : 1975-1999
Fabricant : fabricant non renseigné
Domaines : Informatique et Communication
Sous-domaines : Logiciel
Organisme : ACONIT
Ville : Lannion (Côtes-d'Armor)
Modèle :
Matériaux : Carton, Papier

Description

Le dossier PILOTE-METACOMP comprend une article technique de la revue R.I.R.O. (Revue française d'Informatique et de Recherche Opérationnelle), point de départ du projet, et l'ensemble des documents de réalisation de l'étude : listing des compilations Fortran, "carnet de notes" contenant toute la documentation du projet...

Vocabulaire :

Un compilateur est l'outils logiciel qui permet de transformer le texte en clair d'un logiciel ("code source") en un programme exécutable par un ordinateur ("code objet" puis "code exécutable").

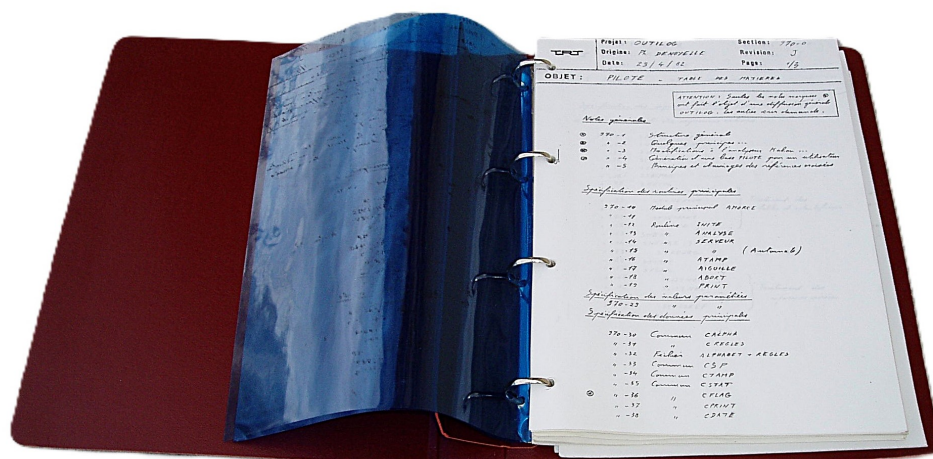
- Un compilateur s'appuie sur une "grammaire" pour analyser un langage et déclencher la génération du code.
- Un métacompilateur prend en charge une grammaire, écrite dans un langage source spécifique, et la traite pour créer le compilateur correspondant.

Utilisation

Le projet PILOTE- METACOMP a été réalisé dans le cadre d'une entreprise de télécommunications. Il a permis aux services logiciels de développer un compilateur (langage de spécification LDS vers assembleur Z80) et plusieurs analyseurs de langages RHM (Relation Homme Machine). Cet ensemble est un document rare est unique qui détaille les éléments techniques nécessaires à l'avancement du projet.

Le point de départ est un article paru en 1969 dans la R.I.R.O. Cet article - précieusement conservé - a servi de base à la réalisation de l'analyseur syntaxique et lexicographique PILOTE en 1981.

Une difficulté d'utilisation de l'analyseur est la création manuelle de la table représentant la grammaire du langage analysé. Ceci a conduit à développer immédiatement un premier compilateur METACOMP, précisément chargé de créer ces tables.



PILOTE —
— METACOMP

AC 18598

ACONIT
18600 (A)

set def [-.pilote]
\$ dyr

Directory DRD3:[DENODIR.PILOTE]

ABORT.FOR;19	5	5-APR-1983	15:50	(RWED,RE,RE,R)
ABORT.OBJ;24	4	5-APR-1983	15:50	(RWED,RE,RE,R)
AIGUILLE.FOR;23	6	8-APR-1982	10:55	(RWED,RE,RE,R)
AIGUILLE.OBJ;28	5	8-APR-1982	10:55	(RWED,RE,RE,R)
AMORCE.FOR;30	15	29-MAR-1982	16:34	(RWED,RE,RE,R)
AMORCE.OBJ;45	9	29-MAR-1982	16:39	(RWED,RE,RE,R)
ANALYSE.FOR;23	17	29-JUN-1982	12:09	(RWED,RE,RE,R)
ANALYSE.OBJ;25	7	29-JUN-1982	12:10	(RWED,RE,RE,R)
ATAMP.FOR;11	5	30-AUG-1983	08:49	(RWED,RE,RE,R)
ATAMP.OBJ;25	3	30-AUG-1983	08:49	(RWED,RE,RE,R)
CALPHA.INC;12	1	9-NOV-1981	10:01	(RWED,RE,RE,R)
CARBRE.INC;6	1	1-DEC-1981	09:00	(RWED,RE,RE,R)
CDATE.INC;2	1	23-DEC-1981	08:25	(RWED,RE,RE,R)
CELEMENT.INC;8	1	30-NOV-1981	09:20	(RWED,RE,RE,R)
CFLAG.INC;14	1	21-JAN-1982	14:42	(RWED,RE,RE,R)
CICROSREF.INC;7	1	15-JAN-1982	09:27	(RWED,RE,RE,R)
CPRINT.INC;9	1	13-JAN-1982	09:40	(RWED,RE,RE,R)
CREFS.INC;1	1	29-DEC-1981	11:52	(RWED,RE,RE,R)
CREGLES.INC;7	4	18-DEC-1981	12:08	(RWED,RE,RE,R)
CROSREF.FOR;56	14	5-APR-1982	16:41	(RWED,RE,RE,R)
CROSREF.OBJ;62	10	5-APR-1982	16:42	(RWED,RE,RE,R)
CSP.INC;1	1	5-NOV-1981	13:40	(RWED,RE,RE,R)
CSTAT.INC;3	1	7-JAN-1982	14:22	(RWED,RE,RE,R)
CTAMP.INC;2	1	6-NOV-1981	08:43	(RWED,RE,RE,R)
CTIDENT.INC;2	1	30-NOV-1981	09:46	(RWED,RE,RE,R)
ELEMENTS.FOR;10	10	8-APR-1983	15:38	(RWED,RE,RE,R)
ELEMENTS.OBJ;22	7	8-APR-1983	15:39	(RWED,RE,RE,R)
GENEPIL.COM;2	14	16-FEB-1982	14:03	(RWED,RE,RE,R)
INITA.FOR;4	2	27-JAN-1982	15:59	(RWED,RE,RE,R)
INITA.OBJ;6	2	27-JAN-1982	16:08	(RWED,RE,RE,R)
LISTING.COM;1	1	31-DEC-1981	10:47	(RWED,RE,RE,R)
MODIFS.DOC;1	2	8-SEP-1983	14:03	(RWED,RWED,RE,RE)
PARAMETER.INC;39	2	19-JAN-1982	11:52	(RWED,RE,RE,R)
PILOTE.COM;7	1	17-JUL-1984	09:46	(RWED,RE,RE,R)
PILOTE.EXE;32	44	30-AUG-1983	08:50	(RWED,RE,RE,RE)
PILOTE.NEW;2	7	22-SEP-1983	10:08	(RWED,RWED,RE,RE)
REPLACE.COM;46	1	15-JAN-1982	09:30	(RWED,RE,RE,R)
SARBRE.FOR;23	16	14-APR-1982	11:56	(RWED,RE,RE,R)
SARBRE.OBJ;37	10	14-APR-1982	11:56	(RWED,RE,RE,R)
SERVEUR.FOR;101	11	29-JUN-1982	15:42	(RWED,RE,RE,R)
SERVEUR.OBJ;106	5	29-JUN-1982	15:43	(RWED,RE,RE,R)
SHEX.FOR;19	9	18-MAY-1983	11:21	(RWED,RE,RE,R)
SHEX.OBJ;20	5	18-MAY-1983	11:23	(RWED,RE,RE,R)
SPRINT.FOR;57	10	2-JUL-1982	10:41	(RWED,RE,RE,R)
SPRINT.OBJ;60	7	2-JUL-1982	10:42	(RWED,RE,RE,R)
STREORG.FOR;1	8	23-DEC-1981	14:15	(RWED,RE,RE,R)
STREORG.OBJ;9	4	19-JAN-1982	11:58	(RWED,RE,RE,R)

Total of 47 files, 294 blocks.

\$

```
set def [-.metacomp]
$ dyr
```

Directory DRD3:[DENODIR.METACOMP]

ALPHAS.FOR;5	7	9-MAR-1982	09:57	(RWED,RE,RE,R)
ALPHAS.OBJ;6	4	3-JAN-1985	17:21	(RWED,RE,RE,R)
CALPHOBJ.INC;8	1	21-JAN-1982	14:39	(RWED,RE,RE,R)
CANVAS.FOR;2	2	18-DEC-1981	17:24	(RWED,RE,RE,R)
CCLASSE.INC;2	1	18-JAN-1982	16:32	(RWED,RE,RE,R)
CFLAGOBJ.INC;6	1	25-JAN-1982	13:49	(RWED,RE,RE,R)
CREGLOBJ.INC;3	1	14-DEC-1981	12:10	(RWED,RE,RE,R)
DBOUCLE.FOR;7	9	9-MAR-1982	10:08	(RWED,RE,RE,R)
DBOUCLE.OBJ;8	2	3-JAN-1985	17:22	(RWED,RE,RE,R)
GESUCALT.FOR;9	14	9-MAR-1982	11:11	(RWED,RE,RE,R)
GESUCALT.OBJ;7	4	3-JAN-1985	17:22	(RWED,RE,RE,R)
INITA.FOR;23	8	9-MAR-1982	11:13	(RWED,RE,RE,R)
INITA.OBJ;33	5	3-JAN-1985	17:22	(RWED,RE,RE,R)
METACOMP.COM;16	1	17-JUL-1984	09:43	(RWED,RE,RE,R)
METACOMP.DAT;33	11	3-JAN-1985	17:28	(RWED,RE,RE,R)
METACOMP.EXE;160	49	3-JAN-1985	17:25	(RWED,RE,RE,RE)
METACOMP.LIS;6	88	8-APR-1982	12:05	(RWED,RE,RE,R)
METACOMP.SRC;12	15	8-APR-1982	11:01	(RWED,RE,RE,R)
MODIFS.DOC;4	2	15-FEB-1985	14:18	(RWED,RWED,RE,RE)
MODIFS.DOC;3	1	14-FEB-1985	08:23	(RWED,RWED,RE,RE)
OBJETS.FOR;22	9	22-SEP-1983	09:33	(RWED,RE,RE,R)
OBJETS.OBJ;19	6	3-JAN-1985	17:22	(RWED,RE,RE,R)
OUVREGLE.FOR;4	15	9-MAR-1982	10:42	(RWED,RE,RE,R)
OUVREGLE.OBJ;6	8	3-JAN-1985	17:22	(RWED,RE,RE,R)
PARAMETA.INC;14	1	3-JAN-1985	17:12	(RWED,RE,RE,R)
REPLACE.COM;23	1	8-FEB-1982	16:35	(RWED,RE,RE,R)

Total of 26 files, 266 blocks.

\$ ty *.doc,*.com,*.inc,*.for

ACONIT
18600 (C)

AC 18596

Doc → H. Denoyelle
R.T.O.
(2^e année, 1^{er} sem., p. 31-72)

ACONIT
18596

PRINCIPES D'UNE METHODE SYNTAXIQUE D'ECRITURE DE COMPILATEURS

par D. KAHAN et H. DURAS-PROBIAULT,
Ingénieurs à la C.I.T.

Résumé. — Cet article expose certains principes de la méthode utilisée à l'occasion de l'élaboration d'un compilateur expérimental au sein de la C.I.T. Le développement de cette méthode a comporté deux étapes :

— La première étape, commencée au début de 1966 (1), a duré un an environ. Confrontée aux problèmes d'analyse syntaxique, elle s'est traduite par l'élaboration de l'analyseur dont le principe est indiqué au paragraphe 3 et par la mise au point de la technique de mise en forme des symboles des langages de programmation en vue de leur exploitation par l'analyseur. Elle a été complétée par une vérification expérimentale du traitement de l'ALGOL du FORTRAN et d'un large sous-ensemble du PLI.

La deuxième étape, d'une durée à peu près équivalente, a consisté dans l'étude des conditions de réalisation et d'optimisation du code objet à partir des données fournies par l'analyseur. Elle s'est matérialisée par l'élaboration d'un compilateur expérimental qui réalise le FORTRAN II de la C.I.T. 90-40. La comparaison des performances entre le compilateur existant de conception classique et le compilateur expérimental, a été dans l'ensemble nettement à l'avantage de ce dernier.

L'objet de cet exposé est l'aspect syntaxique de cette méthode.

1. FORMALISME POUR LA DESCRIPTION SYNTAXIQUE

Il sera présenté à l'aide d'exemples.

1.1. Considérons les règles syntaxiques suivantes exprimées à l'aide des notations utilisées pour l'ALGOL (B.N.F.)

```
< prog > ::= < instr >
< instr > ::= V = < E > ;
< E > ::= < T > | < T > + < E >
< T > ::= < P > | < P > * < T >
< P > ::= < ( E > > | V | N
```

(1) Sous la direction de M. Starykevitch.

Pour nous citer :

Base de la Mission nationale de sauvegarde et de valorisation du patrimoine scientifique et technique contemporain, PATSTEC, Logiciel : Ensemble Projet de programme du Métacompilateur PILOTE-METACOMP (fabricant non renseigné),
<https://www.patstec.fr/ressources/objets/detail?id=27127>, consulté le 2026-07-27