

The Siemens PEARL Compiler System

Dipl.-Math. H. Schoknecht, Dr. rer. nat. P. Rieder, Karlsruhe

1. Language Subset Implemented

The language subset implemented covers Basic PEARL with the following supplements, all language elements from full PEARL:

- initialization of arrays and structures
- more than three dimensions for arrays
- arbitrary lower dimension bound of arrays (also negative)
- assignments of complete arrays and structures
- multiple assignment is allowed
- operators LWB, UPB also monadic
- bit and character group selection (also on the left side of an assignment)
- with task operations schedule lists are allowed
- ACTIVATE with priority parameter
- Modulo function (REM) is implemented
- String assignments with cutting off (with output of a warning)
- TOFIXED for CHAR (2)
- TOCHAR for FIXED (7)
- B2 format is allowed
- for the following short forms, the corresponding long forms are also permitted:

CHAR	CHARACTER
DECL	DECLARE
DUR	DURATION
IDENT	IDENTICAL
INIT	INITIAL
IRPT	INTERRUPT
PRIO	PRIORITY
PROC	PROCEDURE
SPC	SPECIFY

2. Brief Description of the Compiler Technology Applied

2.1. Integration into the System Software

Considering the later introduction of the PEARL Compiler as a product, from the beginning importance was attached to its full integration into the line of products of the development system for the 300/16 bit computer family (/1/, /2/, /3/). This led to the following objectives:

- Programming of the PEARL compiler in the available programming languages, the ASS 300 assembly language and the MECO 300 syntax analysis language, which the maintenance department is already well acquainted with.
- Generation of the object code GS 300, the representation of the machine language of the 300/16 bit computer family, suitable for further processing through linkage editors and loaders.
- Compilation of the modules from source language libraries into object code libraries conform with the module and library structure required by the available utility programs of the 300/16 bit computer family.
- Possibility of linking the modules produced by the PEARL compiler to other GS 300 modules.
- Executability of the compiler as a background program in a limited partition of 17 Kwords (16 bit word length).
- Executability of the compiler and of the programs compiled by it under the control of the ORG 300 standard operating system.

2.2. Promotion of the Acceptance by the Users

Important criteria for the compiler design also resulted from the objective to promote a positive user attitude towards the new product by means of an efficient implementation. This led to the following demands:

- high compilation speed despite of partition limitations.
- detailed and exact compiler messages, informing on the type and the location of the error discovered.
- high efficiency of the object code generated by means of full use of the instruction set of the 300/16 bit computer family without an additional optimization pass.
- integration of high performance runtime test aids into the PEARL compiler system.
- error recognition, if possible at compile time (thorough type testing)
- differentiated processing of inner and outer events (signals, alarms).

2.3. Characteristics of the Realization

The Siemens PEARL Compiler PC 30 was realized as an 8 pass compiler (see Fig.1)

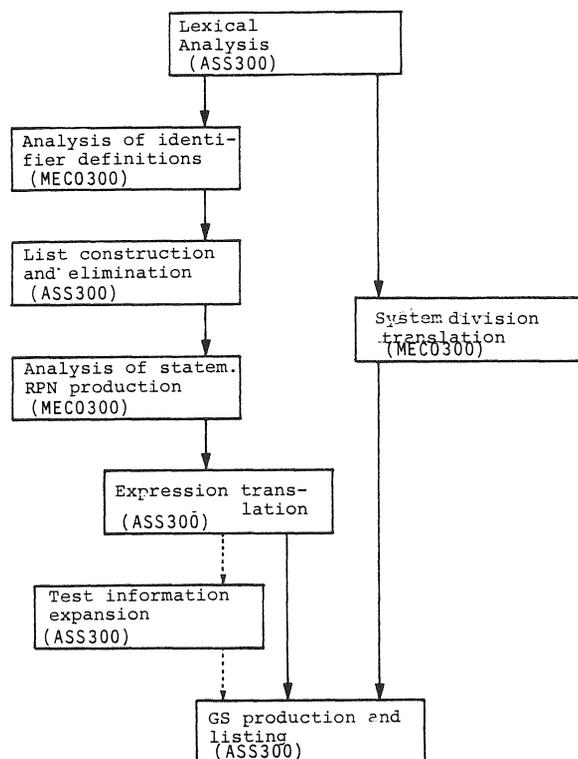


Fig. 1: Passes of the PEARL compiler

and contains, in short, the following characteristics (/4/, /5/, /12/)

- 17 Kwords partition
- lexical analysis by a finite automaton
- division of the syntactic and semantic analysis into four passes:
 - processing of the system division declarations, statements and identifier elimination. The interface between the passes is a common intermediate language, requiring few additional tables. The syntax analysis is implemented according to the syntactic functions method (Glennie syntax machine). Dead ends are avoided by additional bottom-up elements.
- code generation is implemented according to the principle of a stack machine whose first two elements are kept in two register sets. (On principle, the right operand is evaluated before the left one).
- the last pass performs the output of the object code and the listing. If errors are discovered, they are reported in form of a detailed message where they occurred.
- the PEARL source language is directly converted into the object code (300-700 source language lines per minute).

3. Existing Components of the Siemens PEARL Compiler System

3.1. Structure of the Siemens PEARL Compiler system

The structure of the compiler system is to meet the requirements of a high compilation speed and a simultaneous limitation of the available partition. Therefore, the compiler was conceived from the beginning as a multiple pass compiler. Three passes are used for compiling the system division, six for the problem division to output machine code of the 300/16 bit computer family. In the test mode of the compiler, a seventh pass, preparing the test aid information, is executed on compilation of the problem division. The first and

the last compilation steps are identical for system division and problem division. Therefore, the compilation is performed in 8 passes (see Fig. 1, /6/, /7/).

3.2. Integration into a compilation system

To produce and execute PEARL programs, several other programs are required apart from the compiler (see Fig. 2). These programs, however, are not PEARL specific but generally applicable utility programs of the 300/16 bit computer family, namely,

- the text editor (MEDIS) for writing and correcting of PEARL source texts.
- the linkage editor (BD 30) for linking PEARL modules together and for linking them to modules of the PEARL run time system or sometimes to modules not written in PEARL, e.g. to assembler procedures.
- the loader of a (standard) operating system for loading of the GS modules produced by the compiler and the production of address references beyond module boundaries.

The entire production path can also be controlled by service masks of the TESEUS software development system.

To execute PEARL programs, the PEARL run time system is necessary apart from a suitable operating system. It contains, in form of procedures, all these instruction sequences and data which, due to their length, are not directly inserted in the code generated by the compiler, but which are addressed via procedure calls.

Most procedures of the run time system are reentrant and therefore, they must be kept in memory only once. To facilitate handling, most of them were collected in the following eight compound modules:

- elementary routines as e.g. routines for block entry, registration of a signal reaction, array addressing etc.
- mathematical functions such as sine, cosine etc.

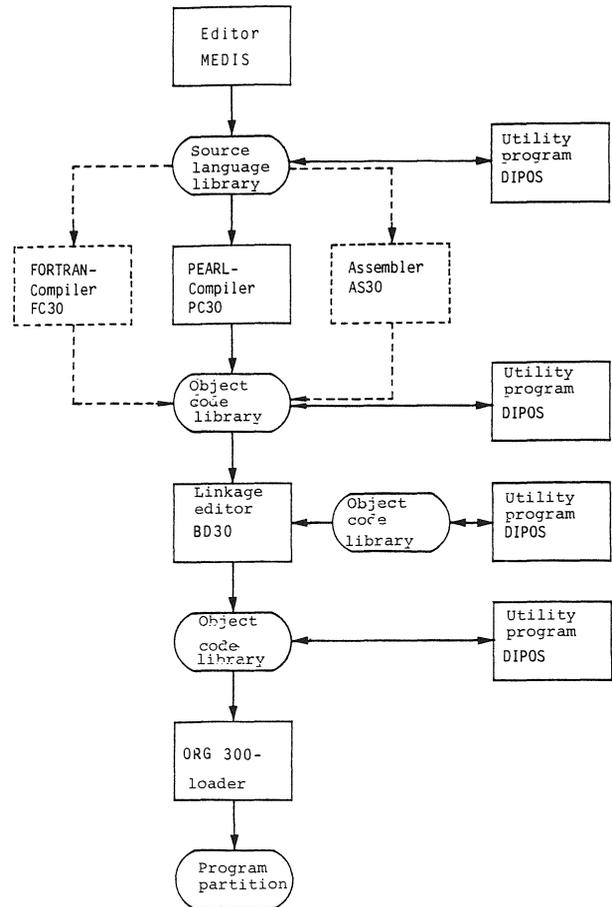


Fig. 2: Production path for PEARL programs

- kernel routines for input and output, especially in binary, unformatted form.
- routines for positioning during input and output.
- routines for realization of the GET statement
- routines for realization of the PUT statement
- routines for input and output into files.

These compound modules can be either linked to PEARL programs or loaded as common code.

Besides the compound modules, there are about 40 reentrant small driver routines which generally are only linked to the task requiring them. For the few non-reentrant run time routines (e.g. the task start routine), the only possibility to make them available is by linking them to the calling task. For interrupt servicing about 120 standard signals divided into 4 reaction classes and 8 different error classes are at the user's disposal.

3.3. Test aids

The existing high-performance run time test aids make use of the information prepared during the test pass and perform the following dialogue-controlled functions:

- listing the numbers of the lines executed
- lines numbering for the error location in the case of a run time error
- unconditional stops at eligible line beginnings
- stops at eligible line beginnings, depending on the value of a variable
- testing and changing of the value of variables.

In the source language listing, the error messages of the compiler are output at the place where they appeared.

4. Host Computers and Target Computers

There is no distinction between host computers and target computers, since cross-compilation is not necessary.

Host computers and target computers may be:

Siemens 330, R10, R20, R30, R40

Configuration for compilation

- console device
- printer
- 17 Kwords partition for the compiler
- 195 Kwords minimum swapping area on disc

Configuration for target computers

- standard peripherals and process peripherals as applicable (in particular console device, printer, disc, graphic CRT terminal, paper tape, process signal interface).

5. Form of Delivery, Training Material, User Manuals, Maintenance Services

Form of Delivery

The PEARL compiler together with the library is delivered on a disc as a segmented program ready for loading.

Training Material

Siemens offers a two-weeks training-course on the language PEARL.

User Manuals

A user manual as well as a short description for the experienced user are available order number (P71100-D3010-X-X-35). Besides the language description these manuals include the directions for use of the compiler and a detailed error and signal description.

Maintenance Services

The customer has a 12 month warranty on the functioning of the compiler system.

6. References and Applications

The first PEARL compiler system was released in January 1978 (/8/).

Since then, three further product releases were realized. They had become necessary due to our field experience. They cover the enhancements according to the users' suggestions (/9/, /10/, /11/).

References, state 9/81

(The PC 30 has been available since 1978)

Industry:

OBAG	Regensburg
Bayer AG	Krefeld
MBB	Ottobrunn
SDR	Stuttgart
GEW	Köln
Berufsförderungs-	
werk	Heidelberg
Battelle	Frankfurt
BWB	Eckernförde

Verbundwerke
 Haus Aden Oberaden
 DFVLR Oberpfaffenhofen
 NDR Hamburg
 EWAG Nürnberg
 Raubach & Co Freiburg

Colleges:

Stuttgart, Berlin, Karlsruhe
 Göttingen, Darmstadt, Dortmund

Internal References:

Application in several areas

7. Literature References

- /1/ Rieder, P.: Effiziente PEARL-Implementierung für den PR 330, Informatik Fachberichte, Band 7, S. 173-183, Springer-Verlag, Berlin/Heidelberg/New York 1977
- /2/ Degelow, L., Gottwald, H.-J., Schoknecht, H.: Stand der PEARL Entwicklung. Tagungsbericht der 8. Jahrestagung des Siemens Anwenderkreises, S. 73-89, Fachhochschule Dortmund, 1977
- /3/ Schoknecht, H.: PEARL 300: Kompiliersystem und Verwendung für Echtzeitaufgaben. Tagungsbericht der 9. Jahrestagung des Siemens Anwenderkreises, Bericht KfK 2642, S.161-175, Ges. für Kernforschung mbH, Karlsruhe, 1978
- /4/ Gottwald, H.-J., Schoknecht, H.: PEARL, eine leistungsfähige Echtzeit-Programmiersprache. Regelungstechnik S. 23-27 (1978)
- /5/ Dorn, M., Wenzel, T.: Prozeßsprache PEARL 300 für die Siemens-Systeme 300, Siemens-Zeitschrift, Band 52, S. 23-27 (1978).
- /6/ Siemens Erlangen (1980), Übersetzungsprogramme: PEARL 300. Best.Nr.: E-36/2205.
- /7/ Siemens Karlsruhe (1980), PC 30/ PEARL 300: Compiler für PEARL, Programmbeschreibung. Best.Nr.: P71100-D3010-X-X-35.
- /8/ Bamberger, K.-F.: Das PEARL-Kompiliersystem für die Siemens 300-16 Bit. PEARL-Rundschau, Band 1, Nr. 1, S. 49-64 (1980).
- /9/ Struhulla, D.: Erfahrungen mit BASIC-PEARL im Projekt Netzleitstelle Deggen-dorf. PEARL-Rundschau, Band 1, Nr. 2, S. 13-22 (1980).
- /10/ Weber, H.: Einsatz von PEARL bei der Software-Entwicklung für eine Fließ-Schließplatz-Automatisierung. PEARL-Rundschau, Band 1, Nr. 4, S. 26-31, (1980).
- /11/ Mayr, U.: Funktion und Aufgaben der Netzleitstelle Deggen-dorf, PEARL-Rundschau, Band 1, Nr. 2, S. 3-12, (1980).
- /12/ Rieder, P.: Implementierung eines PEARL-Compilers, PDV Entwicklungsberichte E 148, Ges. für Kernforschung mbH, Karlsruhe, 1980.