

# **Charon: a Tool for Code Redocumentation and Re-Engineering**

*Oreste Signore - Mario Loffredo*

*2nd Workshop  
on  
Program Comprehension*

*Capri, Naples-Italy, July 8-9, 1993*



CONSIGLIO  
NAZIONALE  
delle RICERCHE

**SEAL - Software Engineering & Application Laboratory**

**Istituto *CNUCE***

**via S. Maria, 36  
56126 Pisa (Italy)**

# Contents

p **Maintenance**

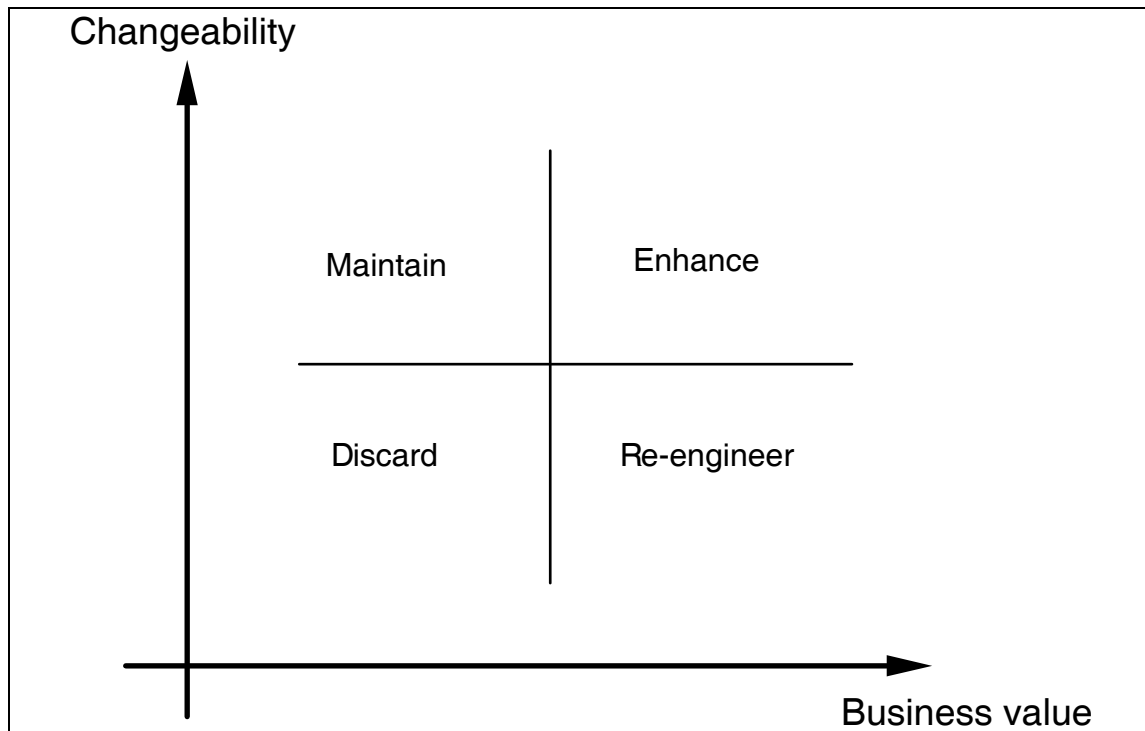
p **Re-engineering & Reverse Engineering**

p ***Charon*: an integration with a CASE tool**

- **Basic issues**
- **ADW**
- **C-TOOL**
- **The ADW files**
- **The import process**
- **A sample**

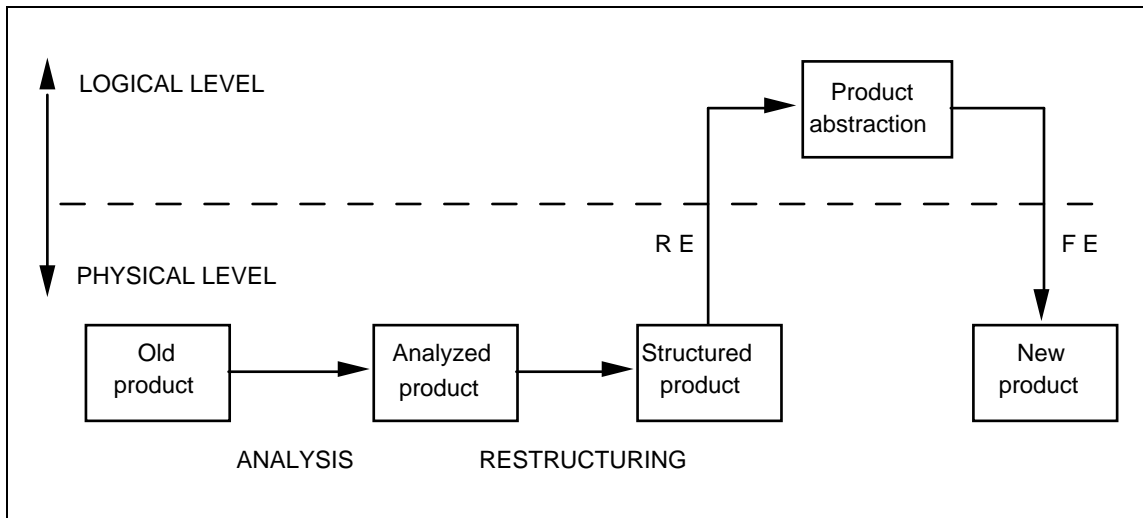
p **Conclusions and possible developments**

# Maintenance



- **Maintenance (corrective, adaptive or perfective):**
  - *up to 95% of EDP departments activity*
  - *we must understand the semantics of the software*
- **CASE tools claimed to be cost and quality effective:**
  - *recover existing software and manage it in a CASE environment*
- **Maintenance interventions may:**
  - *alter the features of the original software*
  - *contribute to their degrade*
  - *make the application system less reliable and maintainable, error prone, difficult and expensive to be modified*
  - *make their documentation out of date*
  - *need to rebuild knowledge before making a change*
- **Effective prototype systems may need to be migrated to more complex or operational environments.**

# The Re-Engineering process

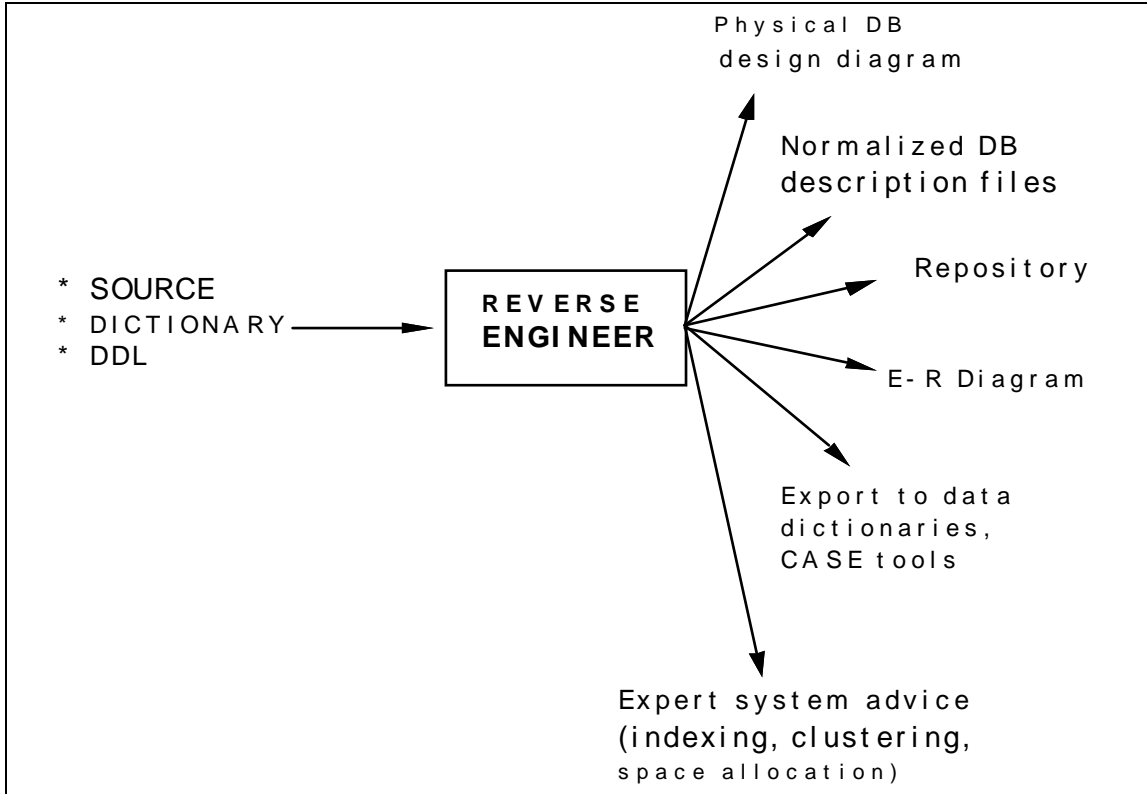


## Objectives of Re-engineering:

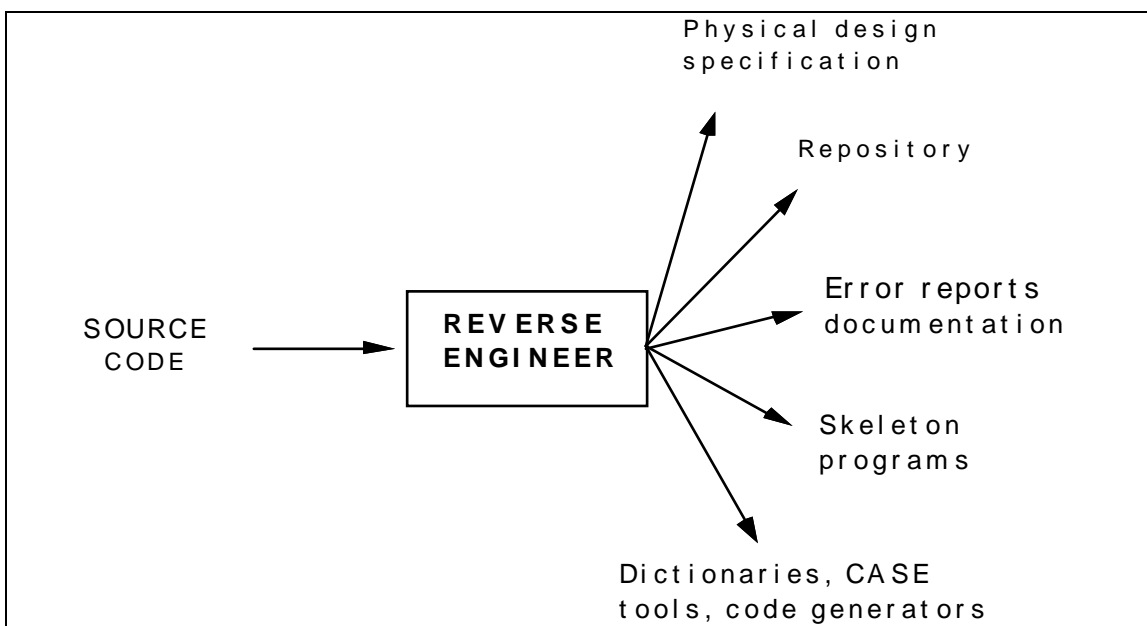
- **Better manage portfolio of existing systems.**
- **Provide automated assistance for maintenance.**
- **Reduce maintenance errors and costs.**
- **Increase productivity of system maintainers.**
- **Make system easier to understand, change, test.**
- **Enable system conversion and migration.**
- **Improve maintenance staff morale.**
- **Enforce adherence to standards.**
- **Improve response to maintenance request.**
- **Protect and extend system life.**
- **Use CASE to support existing systems.**
- **Reuse existing system components.**

# Reverse Engineering...

## p Data...



## p ...and processes





## Related work

- **H. M. Edwards, M. Munro**  
***RECAST: Reverse engineering from COBOL to SSADM Specification (WCRE'93)***
- **D. P. Olshefsky, A. Cole**  
***A prototype System For static and Dynamic Program Understanding (WCRE'93)***  
(PUNDIT: analysis of C source code and production of a set of program views)
- **W. J. Premerlani, M.R. Blaha**  
***An approach for Reverse Engineering of Relational Databases (WCRE'93)***
- **Bachman**  
***Bachman Product Set***  
(Re-engineering of data structures to E-R)
- **Arthur Young Management Consulting**  
***TIBER (Techniques and Instruments to Build Encyclopaedias of Redevelopment engineering)***  
(From COBOL code to IEW)
- **F. Lanubile, P. Maresca, G. Visaggio**  
***An Environment for the Reengineering of Pascal Programs (CSM '91)***  
(From Pascal code to IEW)

## Some basic issues

p **Software quality improvement may require a redevelopment of the application system.**

p **When a software development methodology and CASE tools have been adopted as an enterprise standard:**

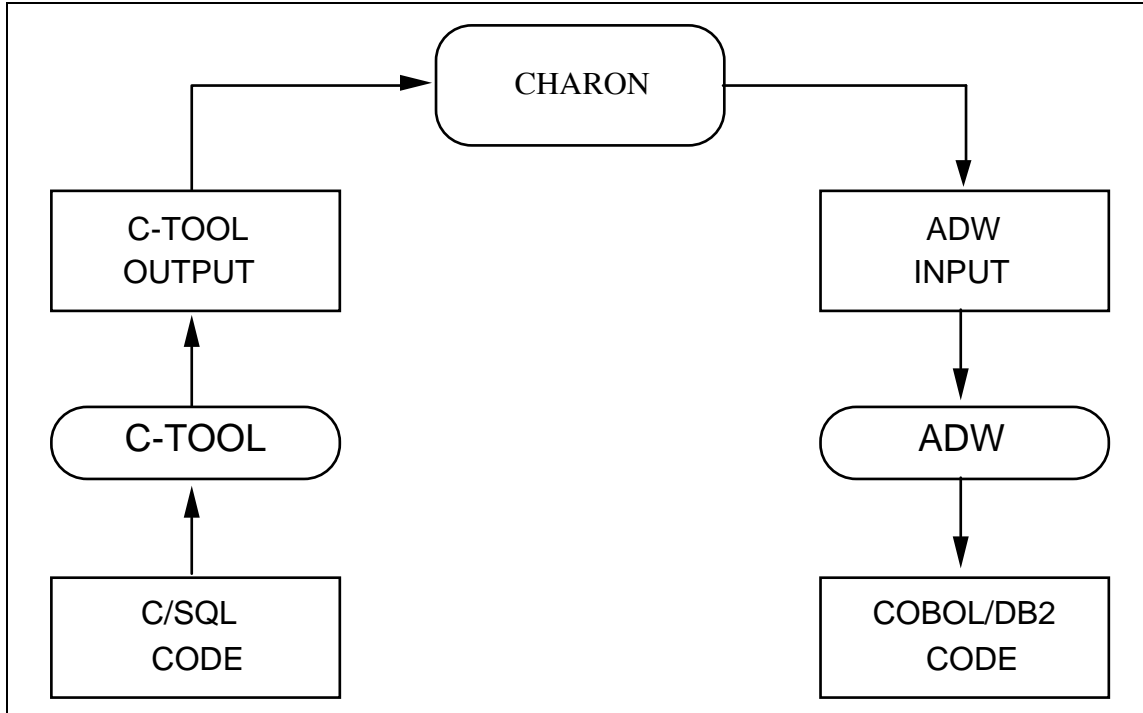
- *recovering the existing software, and documenting it according to these standards may constitute a consistent improvement*
- *understanding the semantics of the original programs is a key point*
- *in large scale projects re-engineering and the adoption of CASE tools can produce relevant advantages, namely consistency, easy maintenance and clean documentation*
- *consistency with the enterprise standards can assure the complete integration of the various subsystems, and reduce the maintenance effort*
- *the maintenance personnel will no more be forced to operate maintenance interventions on the source code, but can operate on higher level specifications, leaving to the CASE tool the burden of the generation of the code*
- *documentation will be kept up to date*

p **An approach: *Charon***

*(we extract from the source code information pertinent to analysis and design phases)*



# CHARON: the objectives



- p **Re-engineering cycle implementation:**
  - from a C MicroSoft language + EXEC SQL program
  - to COBOL code + DB2
  
- p **Reverse engineering: C-TOOL**
  
- p **Forward engineering: ADW COBOL code automatic generator**

# CHARON: the objectives (cont.)

- p **Reconstruct the Database Conceptual Schema**  
*(information pertinent to AWS, while C-TOOL acts at the DWS level)*
  
- p **The designer can afterwards operate directly upon the Conceptual Schema.**  
**The Relational Translator can produce a normalised relational schema.**
  
- p **A bottom-up approach**  
*(from the code to the high level specifications)*
  
- p **A software engineering experiment**

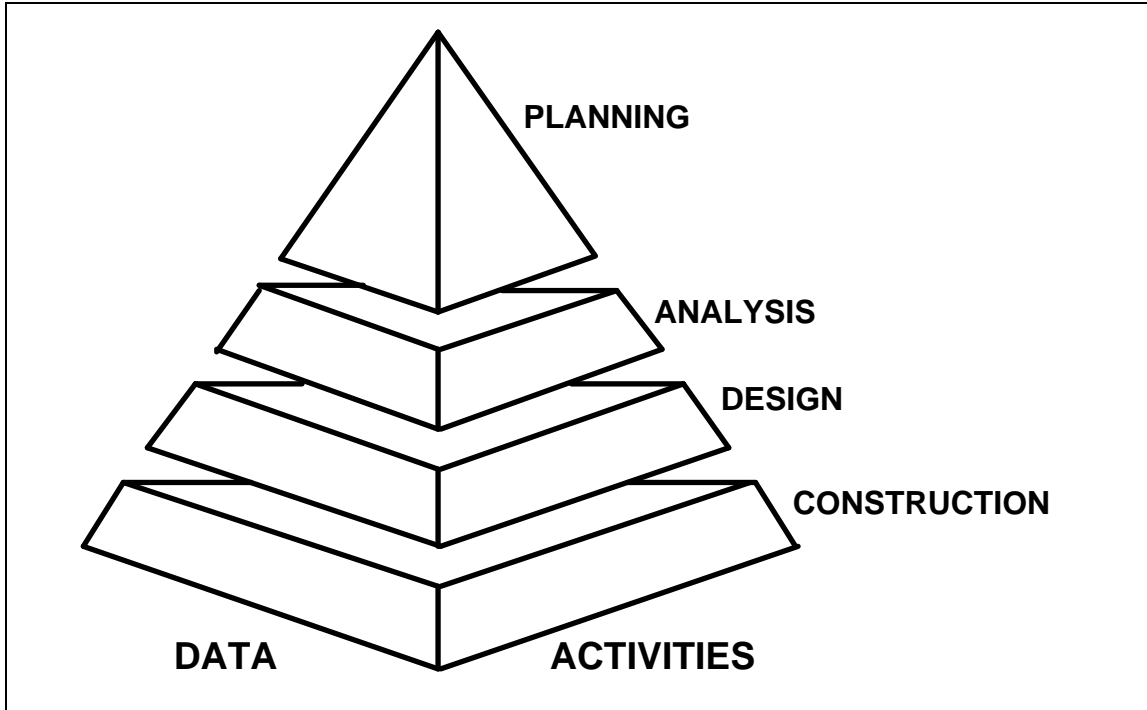
*... but, in addition ...*

**a tool for:**

- **software packages modification**
- **environment conversion**
- **from prototypes to production**

- p **Complete reverse phase executed by C-TOOL**

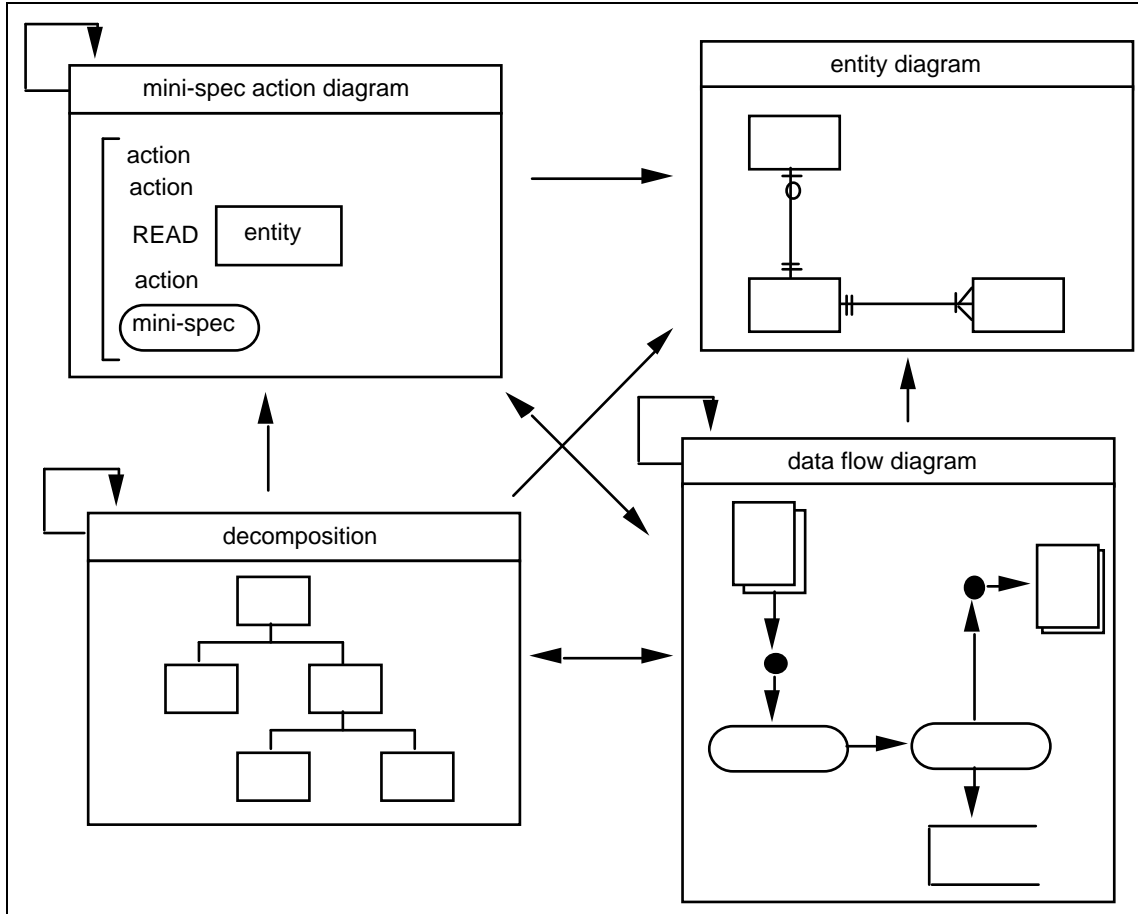
# ADW



p **An implementation of the Martin “pyramid”**

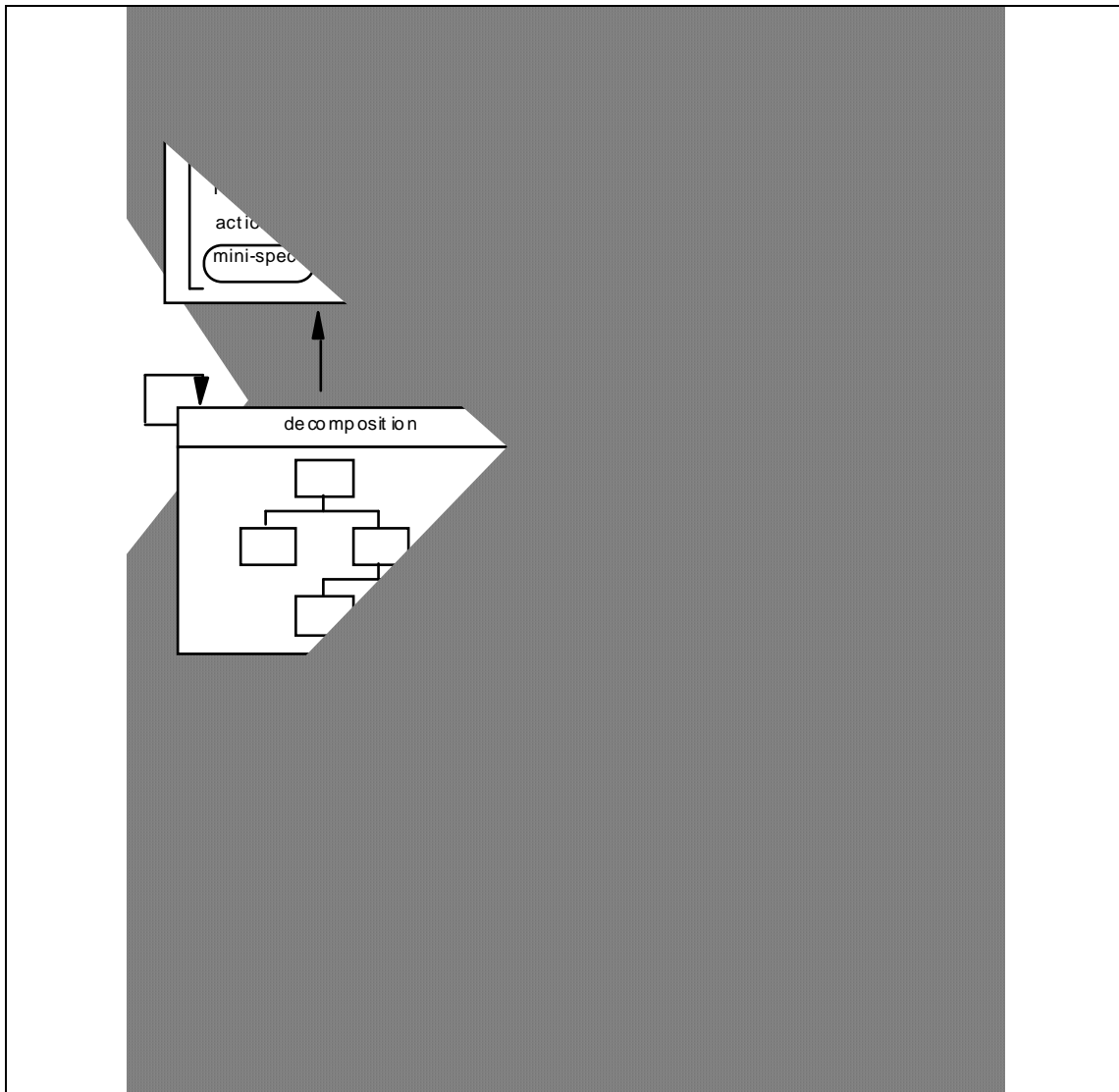
p **A component of the AD/Cycle platform**

# The hyperdiagrams



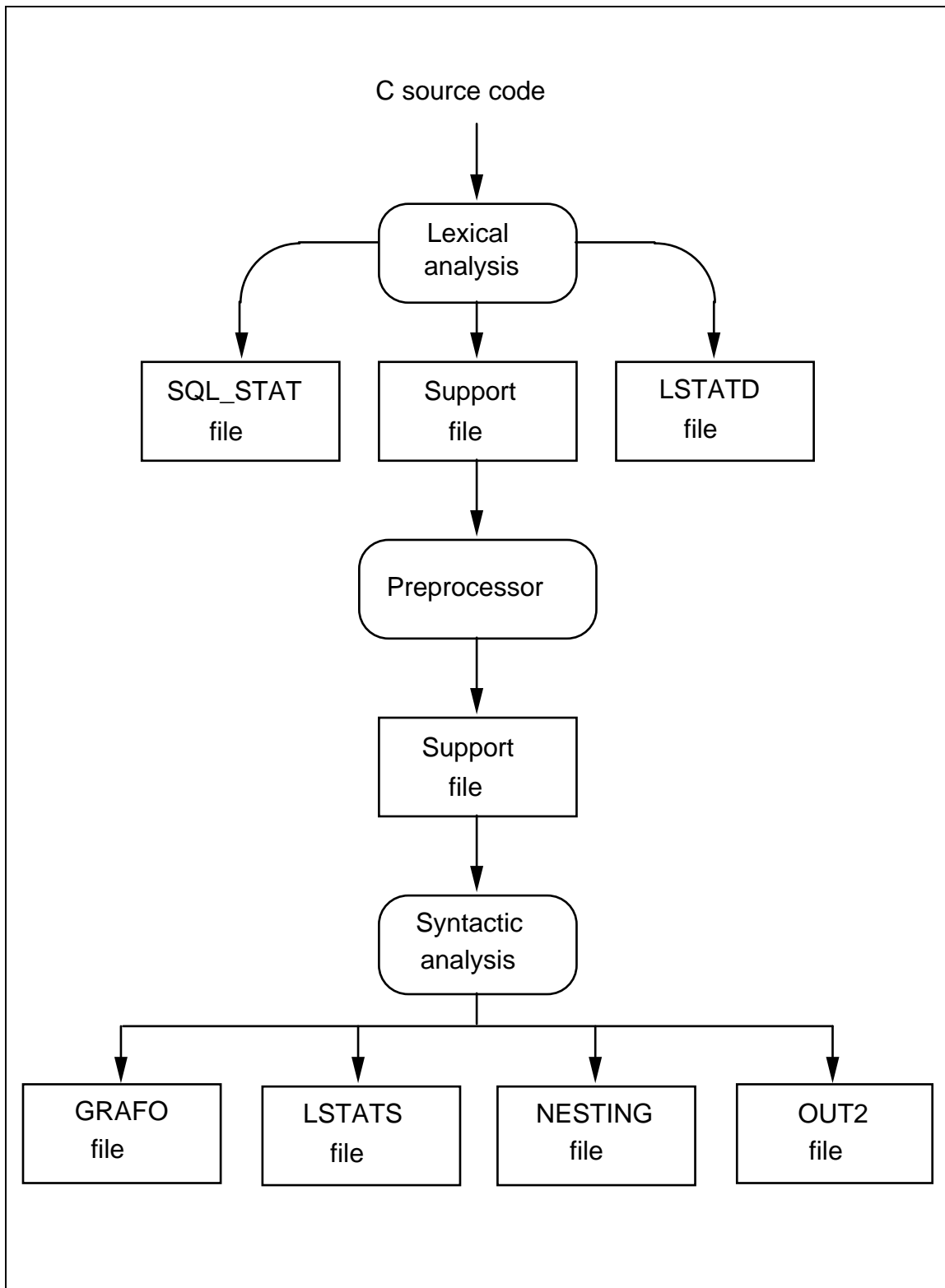
- p **Support the most popular analysis and design techniques (i.e.: DFD, SC, MAD, ERD, RDD, DSD).**
- p **Allow the logical binding among many different representations.**
- p **Are automatically testable and easy to manage.**

# The Encyclopedia



- p **Is a *knowledge base* containing all the project information.**
- p **Is no DBMS based (B-trieve).**
- p **Is integrated with a software component, named *Knowledge Coordinator*, that adopts AI techniques to maintain the diagrams' consistency.**

# C-TOOL: architecture



## The C-TOOL output files

- p ***SQL\_STAT file:***  
includes all the EXEC SQL commands being in the C source code and their positions
  
- p ***OUT2 file:***  
contains the program Call Graph described as a table
  
- p ***NESTING file:***  
includes the program Nesting Tree represented as a list of records
  
- p ***GRAPH file:***  
contains the program Control Flow Graph represented as a list of records

# The files to import in ADW

They can be divided into three groups:

- p ***".EXP" files:***  
including records used to describe all the project information entities as objects, associations and properties
  
- p ***".ENC" files:***  
each one containing the representation of a project module Action Diagram
  
- p ***".MASC" files:***  
each one collecting the masks of SQL DML commands on certain table



## ...the “.EXP” files format...

**OI.EXP - includes the encyclopedia objects representations**

Object instance token	Object type code	Object instance name
-----------------------------	------------------------	----------------------------

**AI.EXP - includes the representations of the associations between the objects involved**

Association instance token	Assoc. type code	From token	To token
----------------------------------	------------------------	---------------	-------------

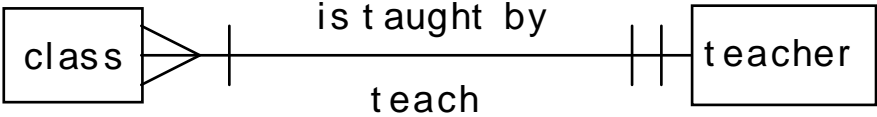
**PI.EXP - contains the information related to the properties of objects and associations**

Subject token	Prop. type code	Rep. no.	Property value
------------------	-----------------------	-------------	-------------------

**TI.EXP - contains the records regarding the long textual properties of objects and associations like definitions and comments**

Subject token	Prop. type code	Rep. no.	Long textual property value
------------------	-----------------------	-------------	-----------------------------------

**..an example of “.EXP” contents...**



***Objects***

10000000003,10007,"teacher" "

10000000004,10007,"class" "

***Associations***

20000000022,20044,10000000004,10000000003

***Short properties***

20000000022,30034,00000,"is taught by" "

20000000022,30037,00000,"teach" "

20000000022,30035,00000,"1" "

20000000022,30036,00000,"1" "

20000000022,30035,00000,"M" "

20000000022,30036,00000,"M" "

***Long textual properties***

10000000004,30076,00001,"A\_group\_of\_students\_which\_study\_" "

10000000004,30076,00002,"the\_same\_arguments" "

## ...the “.ENC” files format...

**\$ADTEXT\$3.00ENGLISH**

**L 0000000For Each Customer Purchase**

**D 0000004Customer&oi000181Using Customer No or Customer Name**

**B 0000000If Customer does not have a Customer No**

**A 0000001Maintain\_Customer\_Data&oi0000LZ**

**C 0000000Else Customer No does exist**

**0000000Check Current Credit Rating**

**B 0000000If Current Credit Rating is > or = 90**

**X30000000Reject Customer Purchase**

**0000000Reason: Credit Past Due**

**C 0000000Else Current Credit Rating is < 90**

**0000000Customer is Valid**

**D 0000005Customer&oi000182Customer Name, Customer Address**

**E 0000000**

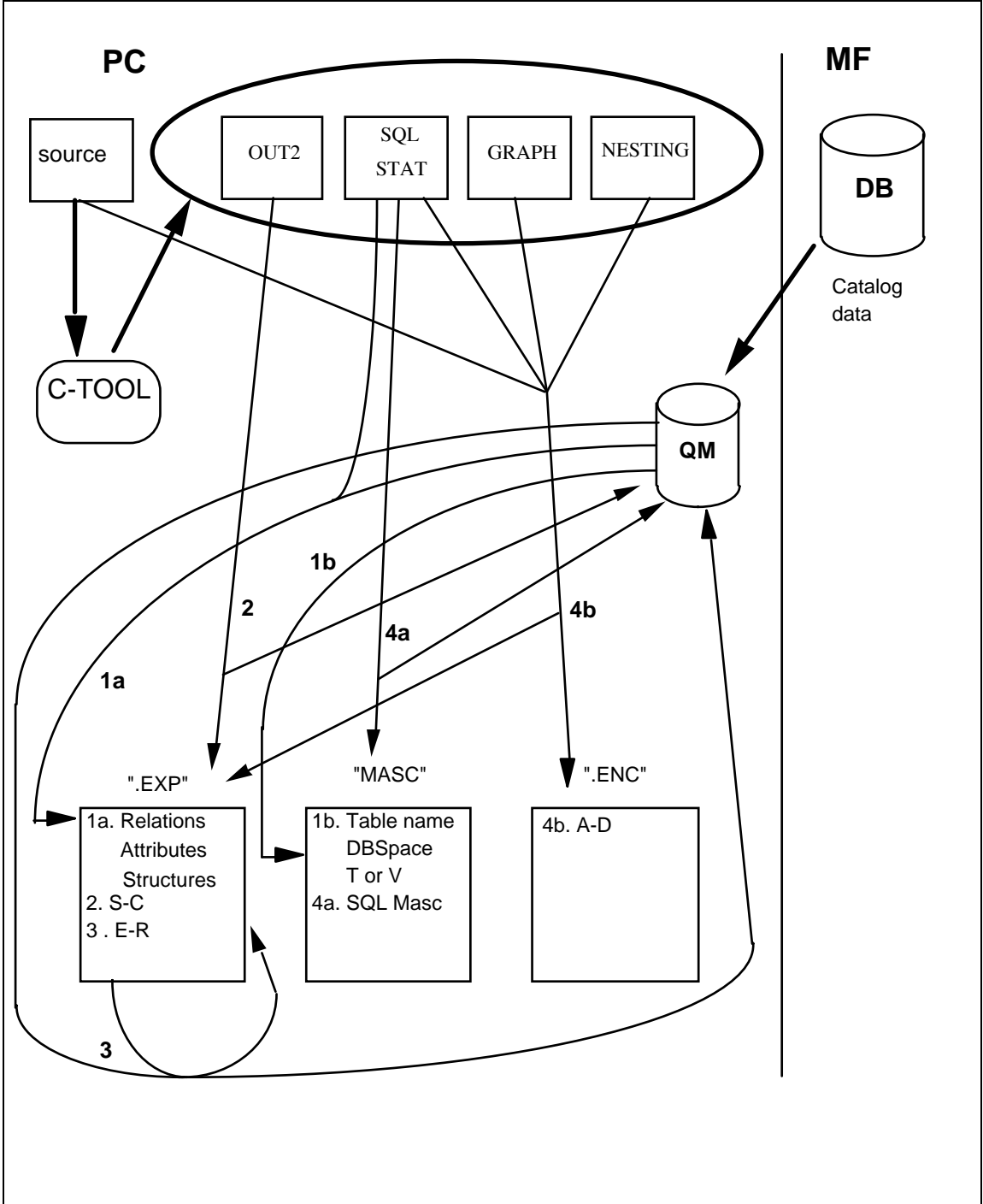
**E 0000000**

**F 0000000**

## ...the "MASC" files format...

```
67,osG00indagini          newuser 06 Feb 922:88am
83,MLosG00A100000indagini
83,MLosG00B100000ospedale.indagini
83,MLosG00C100000ospedaledbsn4
83,MLosG00P100000TV
87,MLosG00R10100000indagini-CURSORO-1      *CURSOR
31,MLosG00R101001B      0000000EXEC SQL
45,MLosG00R101000      0000000DECLARE      c1
33,MLosG00R101001      0000000CURSOR FOR
58,MLosG00R101002B      0000000SELECT      codice_indagine
53,MLosG00R101003      0000000      , tipo_indagine
52,MLosG00R101004      0000000      , numero_volte
54,MLosG00R101005      0000000      , data_richiesta
52,MLosG00R101006      0000000      , espletamento
23,MLosG00R101007E      0000000
45,MLosG00R101008      0000000FETCH      c1
65,MLosG00R101009B      0000000INTO      :osG00-codice_indagine
60,MLosG00R101010      0000000      , :osG00-tipo_indagine
59,MLosG00R101011      0000000      , :osG00-numero_volte
61,MLosG00R101012      0000000      , :osG00-data_richiesta
59,MLosG00R101013      0000000      , :osG00-espletamento
23,MLosG00R101014E      0000000
51,MLosG00R101015      0000000FROM      indagini
63,MLosG00R101016B      0000000WHERE      numero_ammissione =
64,MLosG00R101017      0000000      :osG00-numero_ammissione
40,MLosG00R101018      0000000
23,MLosG00R101019E      0000000
31,MLosG00R101020E      0000000END-EXEC
```

# CHARON: the architecture



# CHARON: the architecture (cont.)

p **Step 1**

- 1a) **Represent the logical relational model and the data Structure of the relational tables**  
*(keys, data types, formats).*
- 1b) **Create and open “MASC” files.**

p **Step 2**

**Reconstruct the Structure Chart.**

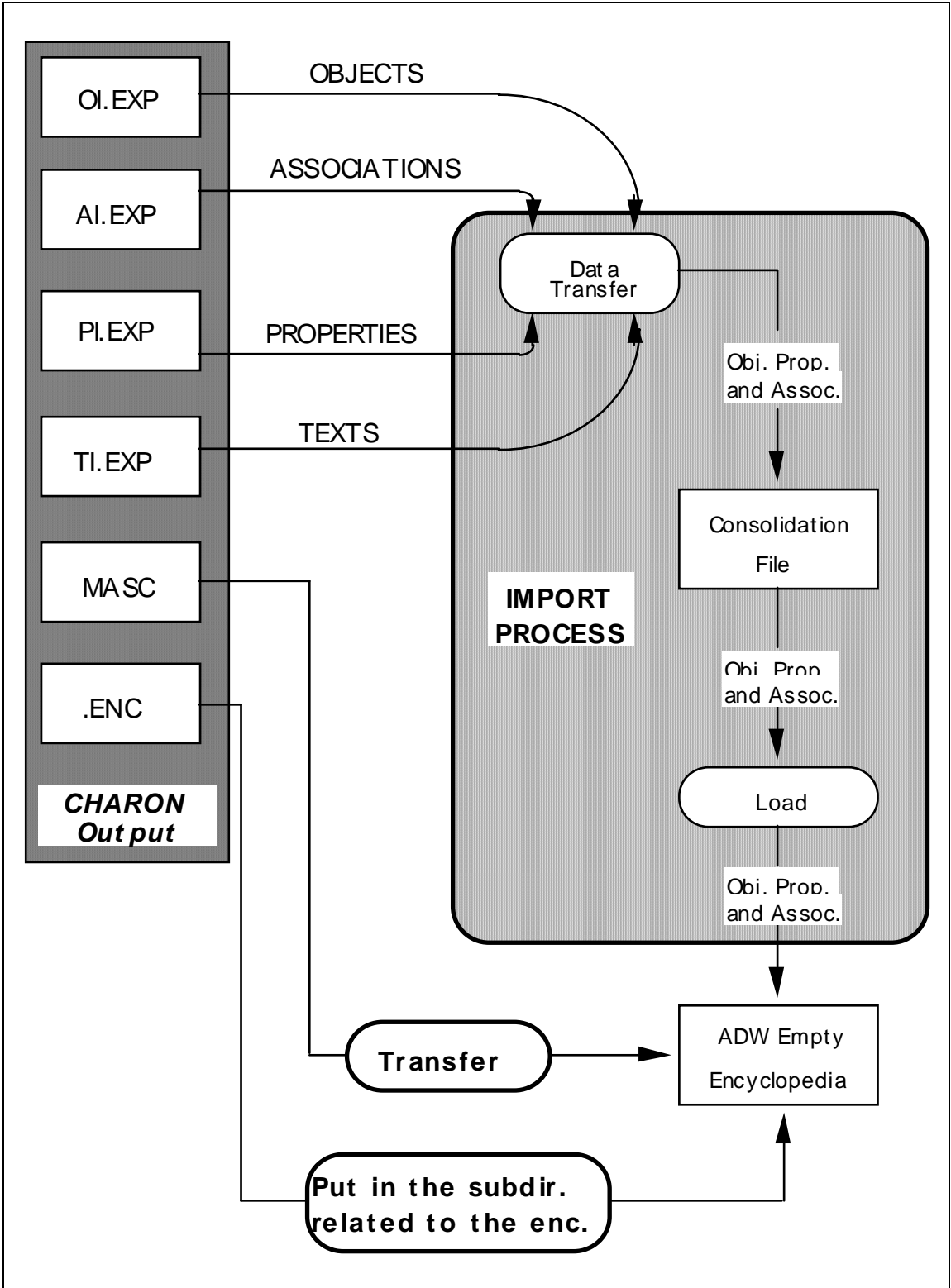
p **Step 3**

**Generate the E-R model**

p **Step 4**

- 4a) **Represent the modules procedural logic, including the accesses to the database and the calls to other modules.**
- 4b) **Transform the EXEC SQL statements into masks to be inserted into “MASC” files.**  
**The statements that are not supported by ADW are inserted as comments.**

# ...importing files...



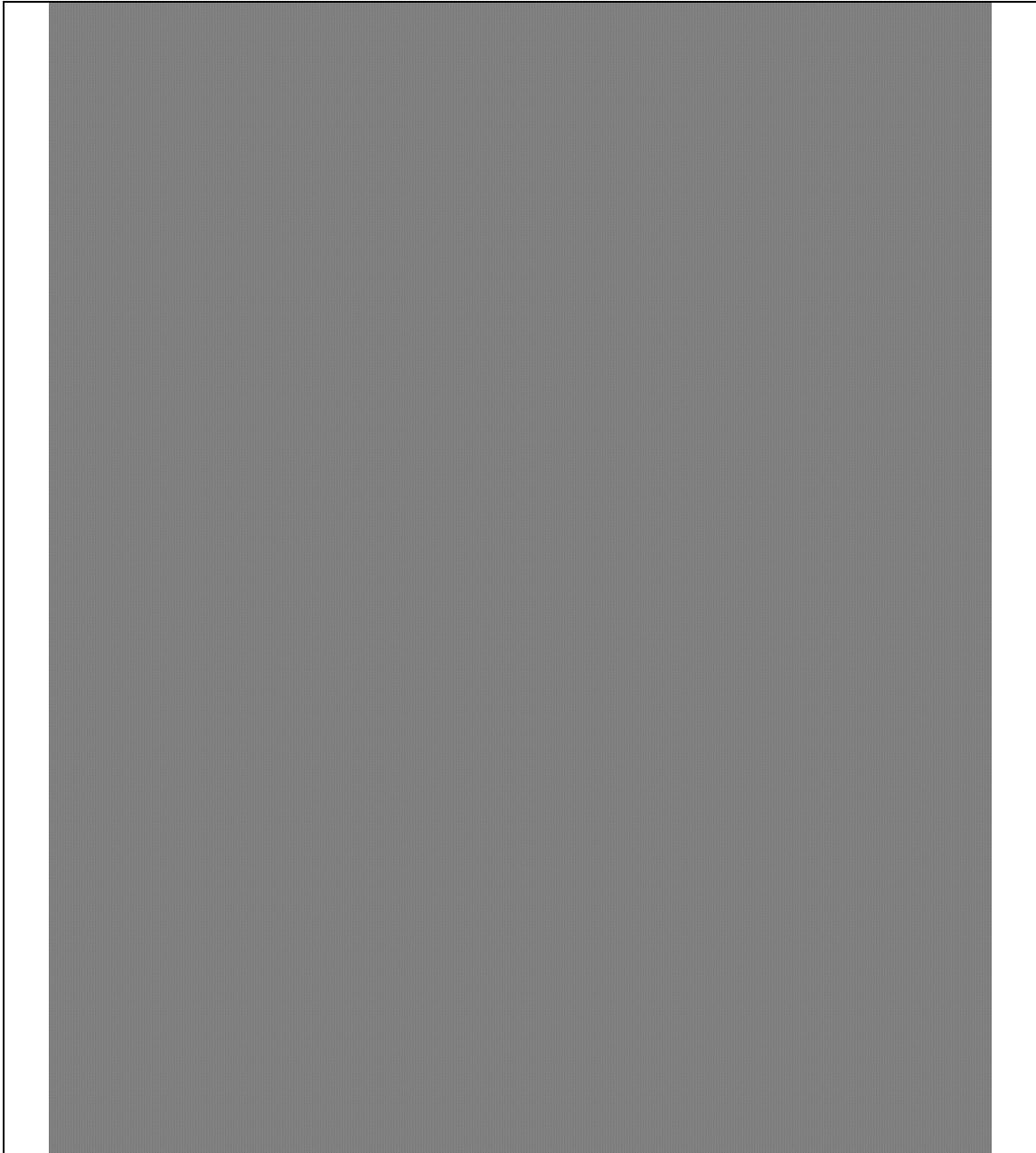
## ...importing files

### A different way of importing per type:

- p **".EXP" files:**  
imported by using the *Encyclopedia Data Transfer* option of the *File* menu of the *Encyclopedia Services* task present in each workstation
  
- p **".ENC" files:**  
moved in the ADW subdirectory associated to the new encyclopedia
  
- p **".MASC" files:**  
transferred through the *Transfer* option of the *File* menu of the *CWS Code Generator* task



## CHARON: achieved targets



p **From a source program to a CASE tool.**

**Benefits:**

- *integration with the enterprise standards*
- *get all the advantages of automated software generation*
- *exhaustive documentation and reports*
- *maintenance at higher level*

# Database Relational Diagram

# Data Structure Diagram

(fundamental table)

# Data Structure Diagram

(associative table)

# Data Type Window

# Entity-Relationship Diagram

# Entity Type Description

# SQL Action Diagram



# Structure Chart

# Module Action Diagram

# Object Details Window

# Object List

# Charon: benefits and future developments

## p **Benefits:**

- *totally automatic*
- *the user must conform to some specific design methodology (e.g. Structured Analysis)*
- *after the reverse engineering phase, we can operate directly on the high level specifications of the software, getting all the benefits claimed by the CASE tools*
- *the user can rely on all of the ADW and supporting database manager report writers in order to produce a textual documentation*
- *with the rebuilding of E-R model, if we wish to make some relevant changes to the database structure, we shall easily act on it and, then, obtain a normalised relational form*

## p **Future developments:**

- *testing the existence of the relationships between relations when examining the SQL commands*
- *creating another ADW diagram, Screen Layout, and producing CICS code for the activation of video maps which correspond to the I/O commands in the C code*
- *enhancing the Structure Chart diagram by considering the recursivity*
- *representing modules formal parameters and converting their C types in COBOL ones in order to allow the generation of the Data Flow Diagram in the Analysis level (AWS)*
- *implementing an enriched user interface*