3P

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence
Memo. No. 60

November 1963

# Recent Improvements in DDT

D. J. Edwards and M. L. Minsky

*This empty page was substituted for a blank page in the original document.*

# RECENT IMPROVEMENTS IN DDT

D. J. Edwards and M. L. Minsky

Massachusetts Institute of Technology
Cambridge, Massachusetts

## ABSTRACT

This paper will report new developments and recent improvements to DDT.

"Window DDT" now will remember undefined symbols and define them on a later command. Using sequence breaks, it can change the contents of memory while a program is running, and the contents of memory can be displayed in symbolic form on the scope.

## INTRODUCTION

The distinction between a debugging system and a programming system is not very clear, especially when the two systems work within the same language. In debugging small MACRO programs using DDT, one often adds substantially to the volume of the original program. At least volumetrically, then, he must be programming.

For very small programs it is clearly more efficient, provided access to the computer is freely available, to write the whole program at the keyboard using DDT, because this eliminates a number of preparation phases each of which has a fixed overhead time. With larger programs, on-line programming with DDT becomes more difficult, because there is more to keep track of in what is already written, and more planning to do.

We have added a few features to DDT that make more convenient the construction and debugging of larger programs. Below we describe these new features, and then discuss what it might take to make a really smooth on-line programming system.

### Use of Undefined Symbols

We have added a second symbol table to DDT. This table lists references to symbols used but not yet defined. When a symbol is defined, the references to it in this table are filled-in and moved over to the regular symbol table. This means that one can refer forward to data or program elements not yet provided for, thus reducing the need for advance planning of storage layouts in full detail. The handling of undefined symbols is completely automatic and does not require any new action on the part of the user. When an undefined symbol is typed-in, the machine acknowledges it by typing an over-strike bar. A special control signal makes the system list the currently undefined symbols; another makes the system define all currently undefined symbols in a consecutive block beginning at the currently open location.

### The WINDOW Feature

The control signal xxxx causes a listing of the program starting at xxxx to appear on the



REPRINTED FROM DECUS PROCEEDINGS
Digital Equipment Computer Users Society
Maynard, Massachusetts

scope. This listing has three columns: location, symbolic contents, octal contents. A few special switches on our console then allow one to move this "window" up and down through the program, or "jump" the window to the effective address of an indicated instruction, thus following transfer paths. This feature serves many of the functions of a complete listing and is quite useful in helping to reduce the amount of mental bookkeeping involved in on-line program-construction. It is also very useful in debugging. The window display runs concurrently with DDT, using sequence-break. One operates DDT normally while the window is up, and it indicates changes as they are made in the program.

Sequence Break DDT Feature

Using single-channel sequence-break, DDT remains available while the object program is being run, so that parameters can be changed or patches inserted without stopping the program. If the program contains a scope display output, this means that one can quickly adjust parameters and the like without writing special provisions for this into the object program itself. Again, no new actions are required on the operator's part.

The present version does not allow for object program typewriter input-output.

Discussion

The three features discussed above seem very useful. They do not make DDT into an ideal machine-language assembly program. We plan to improve the system in a number of respects; the result will be a more-or-less complete on-line macro-assembly system.

a) The programs resulting from DDT are not easily relocatable; this makes it hard to build large programs from separately-written small ones. This is particularly regrettable since almost all the information required for relocation is available to the system at input time. If this relocation information is preserved, it can be used not only to make a relocatable program but also to make the system able to provide good symbolic listings, by suppressing meaningless symbolic interpretations of data quantities.

b) The system is weak because it does not have macros.

We plan to add non-recursive macro definitions to the system, and have it produce a relocatable punch-out. Relocation of program blocks during debugging will be permitted.

The version of DDT with improvements (1), (2) and (3) above will be distributed soon through DECUS. No date is set on the relocatable-macro DDT.

**CS-TR Scanning Project**
**Document Control Form**

Date : 11/30/95

**Report #** AIM-60

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☐ Technical Report (TR) ☒ Technical Memo (TM)
☐ Other:_____

## Document Information

Number of pages: 4 (8-IMAGES)
Not to include DOD forms, printer intstructions, etc... original pages only.

Originals are:
☐ Single-sided or
☒ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☐ Unknown ☒ Other:_____

Check each if included with document:

☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number):_____

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

| Description : | Page Number: |
|---|---|
| IMAGE MAP: (1-4) UNMK'D TITLE, BLANK, 41-42 | |
| (5-8) SCANCONTROL, TRGT'S (3) | |

Scanning Agent Signoff:
Date Received: 11/30/95 Date Scanned: 12/6/95 Date Returned: 12/7/95

Scanning Agent Signature: Michael W. Cook

# Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

**Scanned**

**Date:** 12/6/1995

**M.I.T. Libraries**
**Document Services**