ICOT Technical Memorandum: TM-0057

TM-0057

Writing in a Foreign Language and Programming in Warnier's Methodology

— A Study of Programming Processes —

Akihito Taguchi

April, 1984

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191-5
Telex ICOT J32964

Institute for New Generation Computer Technology

Writing in a Foreign Language and Programming in Warnier's Methodology

---A Study of Programming Processes---

Akihito Taguchi

Institute for New Generation Computer Technology
(ICOT)

ABSTRACT

This paper studies the process of writing (not translating) an article in a foreign language using a screen text editor in a manner analogous to Warnier's programming methodology. Such a computer-aided approach can promote the creativity required for writing.

The author's ultimate goal is to develop an creative programming support environment as an intelligent office automation system using a computer network. Future programming, especially based on logic programming, will increasingly come to resemble writing. Therefore, a study of the writing process, especially in a foreign language, may hold clues for research into future programming processes and programming environments, which will supersede the conventional software-life-cycle approach in software engineering.

1. Introduction

Is it possible that computers and computer networks will contribute to human creative activity and intellectual work, such as programming?

Various intelligent office automation systems, including programming support systems, have been proposed. They, however, are aimed only at reducing the user's burden by undertaking routine work or miscellaneous details, rather than at providing essential creative support for human intellectual work or at promoting creativity and imagination. The Japan's Fifth Generation Computer Systems project is also conducting research into intelligent programming support systems as an application of knowledge information processing systems (KIPS) [Yok1]. As Sugita mentions [Sug1], research into information processing, especially in AI, has been aimed at the replacement by automation of human intelligent functions, rather than at the promotion of human creative activities. As he also mentions, the KIPS of the Fifth Generation Computer Systems project would rather focus on developing computers as tools to aid in thinking or in the setting forth of hypotheses, and such computerized tools will replace old ones, such as paper, scissors, paste and fountain-pen.

Some cases of amplifying human imagination are known: e.g., the discovery of the universal gravitation through seeing an apple drop off a tree, and the deep comprehension of the parallelism of data-flow mechanism through playing "PACHINKO" game. ("PACHINKO" is the most popular game among adults in Japan. It is similar to pinball game, but allows to put as many balls as possible into play at the same time.) Forty years ago, Wertheimer described and analyzed some processes of productive thinking or discoveries [Wer1]. The following examples are similar cases in which computer-aids amplify human creativity and imagination.

- .Solving solid-geometry problems by using computer graphics
- .Setting forth hypotheses by using computer simulations
- .Developing software by rapid prototyping
- .Learning actively and voluntarily by playing the LOGO turtle [Pap1]

The author's ultimate goal is to develop a truly intelligent office automation system using computer network systems, especially as related to creative programming support environments [Tag2, Tag3]. There are a number of ill-structured problems for which the conventional programming process in software engineering, based on the software life-cycle concept, is inadequate, and so "many of the methodologies and tools developed in software engineering are of little use [Fis1]." Scott and Scherlis also claim that it is necessary to develop a sound understanding of the programming process and to discover the principles that can be embodied in the future programming tools [Sch1].

This paper studies an experimental experience of writing (not translating) an article in a foreign language using a screen text editor, and the effect of computer aids on the writing process as a creative activity. As Fischer says [Fis1], the design of complex software has much in common with other creative activities (like writing and so on). Moreover, future programming, especially based on logic programming, will increasingly come to resemble writing. Therefore, a fundamental study of the writing process, especially in a foreign language, may also hold clues for the direction of future research into the essential nature of programming processes and into programming support environments, which will supersede the conventional software life-cycle approach.

2. Thinking while Writing

Parenthetically, W.A. Wulf says [Wul1], "A programming language has at least three goals.

. It is a design tool.

. It is a vehicle for human communication.

. It is a vehicle for instructing a computer." This means that programming (not coding) involves not only instructing a computer but also creative thinking. Writing an article is, in itself, a creative activity as well. Writing is not only a vehicle for communicating one's thought but also a means of developing it by using various knowledge bases, such as dictionaries, encyclopedias, books, research papers and the findings of investigations and experiments.

Certain Japanese researchers, especially in social sciences, use methods, called "Card system of Kyoto University" or "KOZANE", for researching and composing articles. (KOZANE originally means a little piece of iron in Japanese. A Japanese ancient armor suit was made of a lot of KOZANEs, so it was strong and flexible.) In both of these methods, basically, researchers compose their articles and set forth their hypotheses by synthesizing flashed ideas and individual data written down on KOZANEs or Cards. It should be remembered that J.Kawakita says in explaining his KJ method, "Let data talk." (KJ method is something like brain-storming.); Data stimulates the creativity and imagination.

3. Writing Processes and Programming Processes

3.1 Writing in a Foreign Language

A language other than one's native language is as constraining and inconvenient as a programming language. As the latter regulates the structure and algorithm of programs, the former regulates one's thought with its characteristic logic. For example, Japanese (the author's native language) and English each have their own characteristic logics. For instance, English is a positional language, i.e., the positions of sentences and paragraphs, in themselves, convey particular meanings. An English article is constructed primarily by building paragraphs. On the contrary, a Japanese article is composed by tying individual sentences together with meaningful conjunctions. Therefore, tools and techniques helpful to the process of writing in a foreign language may also be so to the programming process.

Fig.1 shows various processes for making (writing or translating) a paper in a foreign language. In process (a), the method mentioned above, using KOZANE or card, is adopted, but, here, computer aid is not well utilized in the thinking phase. Moreover, the translation of an article completely organized in the logic of one's native language to another language (having a different characteristic logic) seems to be quite difficult. The author knows from experience that it is much easier to translate dynamically one language into another, as shown in Fig.1 (b). Process (b), in which there is feed-back between thinking and writing, takes advantage of the interactive computerized tool to aid not only in composing a paper but also in the thinking process itself. It should be noted that, in this scheme, the writing of a paper in a foreign language is divided into two subworks (the translation of memoranda and the creation of the framework of the entire paper). In this, such a scheme bears some resemblance to Warnier's programming methodology, as mentioned below.

3.2 Programming Processes and Software Engineering

Fig.2 shows several programming processes to contrast with writing

processes in Fig.1. Process (b) is advantageous to programming as a creative activity for the same reasons mentioned with respect to writing process (b). Fischer also says [Fis1], "By improving the specifications of a problem the designer gets new ideas about how to solve his problem. A documentation of a program that is usable during the design process can therefore be a driving force for the synthesis of new ideas and implementations." Process (a) is nearest to the conventional software engineering approach, based on the life-cycle concept; process (a), as mentioned with respect to writing process (a), also makes no use of computer aids in the creative phase. Software engineering in the 1970's emphasized the management aspects of software development, such as program-version control or production control based on the software life cycle concept, as Zave says [Zav1], "The conventional approach is well-suited to managerial and organizational needs." Consequently, however, it was not very flexible. Moreover, there are a number of ill-structured problems (vague even to the client or the programmer at the beginning) for which the life-cycle approach is inadequate. Therefore, process (c), called operational approach or rapid-prototyping in an executable specification language, is being experimentally adopted for such problems. Zave refers to the difference between the conventional software-life-cycle approach, pervasive in the field of software engineering, and the new operational approach [Zav1].

3.3 Warnier's Programming Methodology

Process (b), shown in Fig.2, is considered to represent Warnier's programming methodology [War1]. "Mapping" is one of the central concepts of Warnier's programming methodology: a program is completed by mapping individual instructions (operations/sentences/commands) onto a global frame of the program. By dividing programming into two subworks, the design of a program framework (control structure) and the selection of actual instructions, programming becomes more tractable. Various human activities, such as planning and designing, are also performed in a similar way. Fig.3 shows, for example, a typical process of travel planning that bears some relationship to Warnier's programming methodology. The travel planning consists of picking up spots to visit during the travel and scheduling them taking into account geographic conditions.

It is interesting to notice that Warnier's basic idea bears some resemblance to the concept of "separation of logic and control" for logic programming [Kow2]. Logic programming will become more important as future programming. By the way, the conceptual idea of Dijkstra's "structured programming" might be said to bear some relationship to the declarative and procedural semantics of a logic program. Dijkstra says [Dij1], "What can we do to shorten the conceptual gap between the static program text and the corresponding (dynamic) computations?" and "I have stressed the need for systematic sequencing so that the structure of the computations could be reflected in the structure of our programs: in this way we can speak of the joint structuring of programs and computations."

This mapping process in Warnier's methodology is similar to the synthesizing process in writing, mentioned above; here the set of instructions is analogous to the sheaf of KOZANEs or cards used in writing an article. Therefore, a computerized screen editor for Warnier's program diagram would facilitate programming in the same way a screen text editor promotes the writing of articles. Such an editor would truncate, insert, replace and shift the twigs and boughs of a program tree diagram, which correspond respectively to the sentences and paragraphs of an article. An experimental interactive PAD-editing system is proposed [Hae1]. A PAD (Problem Analysis Diagram) is a two-dimensional tree-structured diagram (evolved from Warnier's

diagram) for describing the logic of programs [Fut1]. Using a PAD involves repeatedly breaking down a problem solving procedure (often vague even to the programmer at the beginning) into subprocedures until they can be described clearly. So, PADs are not only tools for representing programs but also for thinking about problems.

4. Perspectives on Future Programming Support Environments

4.1 Future Programming Style

One future principal programming style will be interactive "knowledge programming", based on logic programming. This will involve constructing a model or a program while accessing a number of knowledge bases including program libraries and document files; here knowledge means a broad concept including the conventional concepts of program, data, and database [Tag2]. In other words, future programming will be closer to the construction of a model or a hypothesis, or the writing of a research article. A logic program, such as a PROLOG program, dispenses with the conventional concept of "control flow", and has as right the granularity of individual sentences (clauses) as an article written in a natural language: a PROLOG program allows a non-procedural (declarative) interpretation. Parenthetically, the other new programming styles, such as shown below, will also bear a greater similarity to writing or talk, because data-objects and controls will be both higher-level and more abstract.

- .functional programming [Bac1]
- .object-oriented programming [Hen1]
- .abstract data types
- .generic programming [Are1]
- .executable specification languages or rapid-prototyping [Wil1] or operational approach [Zav1]
- .ULTY SHELL programming language [Kou1]

Therefore, a creative programming support environment, for the future, will have much in common with those for other human creative activities.

4.2 Environments and Tools for Creative Activities

As mentioned in Section 1, intelligent office automation systems and intelligent programming systems will not directly support the intellectual work and creative activity of human beings. Therefore, it is important that tools and environments be natural (friendly and flexible) and indifferent ("sarigenai" in Japanese, as mentions Fuchi, Director of ICOT Research Center) so as not to impede human creativity and imagination. As Professor H.Simon mentioned in his lecture given at Keio University in July 1983, attention to the information will be more valuable and creative than the information itself. Which information to pay attention to would best be entrusted to the creativity and imagination of human beings, although intelligent programming support system, as Scott and Scherlis mention in [Sch1], will gradually be provided with knowledge regarding how to use various kinds of knowledge bases necessary to programming. They also say, "We believe that programming, like other areas of creative endeavor, should not be too heavily shackled by form or method. Tools or vehicles for programming, therefore, must be constructed in such a way that programmers will still feel the freedom to explore in unfettered fashion."

Therefor, in order not to constrain the attention of human users, the increased computing power of new generation personal computers should be used to provide an advanced natural communication model of man/machine interface as well as advanced interface media (e.g.

natural language, voice, graphics, and so on). For example, the mode-less communication model concept of Xerox's STAR is based on a multi-window display and gives users the initiative rather than making them dependent on the caprices of the computer [Smi1]. Users can focus attention on any aspects of the computer's operation. They can pay attention to several pages or documents of interest to them at the same time. As M.McCluhan says, "The medium is the message"; the medium is itself a good stimulant to the human imagination.

For a user's attention to be free, visualization is essential. For example, a desk surface is effective not only for "ordering and finding" but also for "reminding" [Mal1]. A bookshelf can also serve a similar function. "Desk-top" and "icon", introduced in STAR, are useful concepts for visualizing the inside of a computer. Moreover, as shown in Fig.4, a multi-window display, combined with another multiplexing technique "virtual circuit" on networks, can be used to visualize an entire network environment, including various knowledge bases distributed around the network [Tag2]. The multi-window display and the virtual circuit may not be sophisticated enough. They, however, present primitive but essential materials required for realizing basic models of man/machine and man/network interfaces. In future, more advanced materials will be available [Tag2].

5. Conclusion

Future programming processes, particularly in knowledge programming and in logic programming, will bear increasing resemblance to the process of writing an article: it might be essential for future programming environments that a programmer's attention be freely paid to various knowledge bases around the network, through the visualization of an entire network environment. Knowledge-based computerized aids, such as a multi-window screen editor of program diagrams and an online documentation, will directly support the programmer's creativity.

However, such future interactive programming support environments will be more advantageous to others than the conventional programming process (based on the software life-cycle concept). Therefore, the research and development of new programming processes will be needed. For such a research, the profound studies of the processes of any kinds of human intellectual activities, including the writing process and the various existing processes of programming (e.g. Warnier's programming and prototyping, as mentioned in this paper) are required. For example, it will be effective to study the process of "walkthrough" [Mci1, You1]; walkthrough, properly performed, might be regarded as a productive thinking process by a group, and is closer to the brain-storming method than to the inspection or desk-debugging of programs. Moreover, because the process of walkthrough is visible, the observation of walkthrough will provide some suggestions for the research of the programming process and support tools.

And, now, the results of new research into the creativity and intellectual functions of human beings, such as the work being done in cognitive science [Fue1, Nor1, Kit2], will also surely contribute to sound understanding such programming processes.

Last of all, whether the present English paper is awkward or not, the author (Japanese) would like to emphasize that he felt comfortable and unconstrained in the course of writing it according to the process described in Fig.1 (b).

6. Acknowledgment

The author would like to thank Masakazu Kobayashi of Hitachi Ltd.

for the opportunity to read his draft paper on the PAD-system, and to thank Toshio Yokoi, Chief of ICOT Third Laboratory, for giving some cue. He would also like to express his gratitude to Kazuhiro Fuchi, Director of ICOT Research Center, and Kunio Murakami, Chief of ICOT First Laboratory, for providing the opportunity to pursue this work.

REFERENCES

[Abe1] Abelson,H. and diSessa,A.,"Turtle Geometry:The Computer as a Medium for Exploring Mathematics",MIT Press,1981

[Arc1] Archibald,J.L.,et.al.,"Abstract Design and Program Translator; New Tools for Software Design", IBM System Journal Vol.22 No.3, 1983

[Bac1] Backus,J.,"Function Level Computing",IEEE Spectrum, Aug. 1982

[Bou1] Bourne,S.R.,"The UNIX System",ADDISON-WESLEY PUBLISHING COMPANY, 1982

[Che1] Cherry,L.,"Writing Tools",IEEE Trans. on COM. Vol. COM-30 No.1, 1982

[Coh1] Cohen, S.,"SPEAKEASY 'RAKUGO'", 1'st USA-JAPAN Computer Conference Proceedings, Oct. 1972

[Dam1] van Dam,A. and Heyrowitz,N.,"Interactive Editing Systems: The Core of Online Thinking",OAC'82 Digest(AFIPS), April 1982

[Dij1] Dijkstra,E.W.,et.al.,"Structured Programming",Academic Press,1972

[Eng1] Engelbart,D.C.,"Toward High-Performance Knowledge Workers", OAC'82 Digest(AFIPS), April 1982

[Fis1] Fischer,G. and Schneider,M.,"Knowledge-Based Communication Processes in Software Engineering",7th ICSE,Mar. 1984

[Fuc1]* Fuchi,K.,et.al.,"An Invitation to Cognitive Science", (In Japanese),Nippon Hosou Kyoukai,Oct. 1983

[Fut1] Futamura,Y.,et.al.,"Development of Computer Programs by Problem Analysis Diagram(PAD)",5th ICSE,Mar. 1981

[Gol1] Goldberg,A. and Robson,D.,"SMALLTALK-80: The Language and its Implementation",Addison Wesly, 1983

[Ish1] Ishii,S.,"Study of Proofreading Techniques Used at a Japanese Newspaper",Proc. of 28th National Conference of Information Processing Society of Japan,Mar. 1984

[Kam1] Kamibayashi,N.,"Object Oriented User Interface Model",IMAC'82,1982

[Kam2] Kamibayashi,N.,et.al.,"Office Professional Workstation(JSTAR) and Integrated Programming Environment Workstation(1100SIP)"(In Japanese),Information Processing Vol.25 No.2,Feb. 1984

[Kit1]* Kitagawa,T.,Kawakita,J. and Nakayama,N.,"Creative Science", (In Japanese),Chuonkoron-Sha Jan. 1971

[Kit2] Kitagawa,T.,"An Informatical Approach to Knowledge Information Processing Systems",(In Japanese),Fujitsu LTD,Mar. 1983

[Kow1] Kowalski,R.A.,"Logic for Problem Solving",North Holland,1980

[Kow2] Kowalski,R.A.,"Algorithm = Logic + Control",CACM Vol.22 No.7,July 1979

[Mae1] Maezawa,M.,Kobayashi,N.,Saito,K. and Futamura,Y.,"Interactive System for Structured Program Production",7th ICSE,Mar. 1984

[Mac1] Macdonald,A.,"Visual Programming",Datamation Vol.28 No.11,Oct. 1982

[Mal1] Malone,T.W.,"How Do People Organize Their Desks? Implications for the Design of Office Information Systems",Proc. of Conference on Office Information Systems

[McC1] McCracken,D.D. and Jackson,M.A.,"Life cycle concept considered harmful",ACM SIGSOFT Software Eng. Note 7 2,1982

[Mon1] Macdonald,N.H. and Poller,M.F.,"Language Processing Tools: The Writer's Workbench and Extentions",OAC'82 Digest(AFIPS), April 1982

[Nor1] Norman,D.A.,Simon,H.A.,et.al.,"Perspectives on Cognitive Science", Ablex Publishing Corporation,1981

[Pap1] Papert,S.,"Mindstorms; Children,Computers and Powerful Ideas",Basic Book Inc.

[Ren1] Rentsch,T.,"Object Oriented Programming", Sigplan Notices Vol.17 No.9, Sept. 1982

[San1] Sandwell,K.,"Programming in an Interactive Environment: the "LISP" Experience", Computing Surveys Vol.10 No.1, Mar. 1978

[Sch1] Scherlis, W. L. and Scott, D. S.,"First Step Towards Inferential Programming", IFIP'83, Sep. 1983

[Sim1] Simon,H.A.,"The Sciences of the Artificial (second edition)", The MIT Press,1982

[Sim1] Simons,G.L.,"Towards Fifth-Generation Computers", NCC Publications, 1982
[Smi1] Smith,D.C.,et.al.,"Designing the Star User Interface",Byte, April 1982
[Ste1] Stefik,M.,et.al.,"Knowledge Programming in LOOPS",AI MAGAZINE Fall 1983
[Sti1] Stibic,V.,"TOOLS OF THE MIND techniques and methods for intellectual work",NORTH-HOLLAND PUBLISHING COMPANY,1982
[Sug1]* Sugita,S.,"Computers for Techniques of Intellectual Work", (In Japanese), Proc. of 27th National Conference of Information Processing Society of Japan,Oct. 1983
[Sug2] Sugita,S.,"Information Processing in Research Museum --Needs at National Museum of Ethnology--",Information Processing Vol.23 No.3,Mar. 1982
[Tag1] Taguchi,A.,et al,"Configuration and Design Philosophy of Operating-System for Personal Sequential Inference Machine (SIM)",(in Japanese),Proc. of 26th National Conference of Information Processing Society of Japan, Oct. 1983
[Tag2] Taguchi,A.,et.al.,"INI: Internal Network in ICOT and its Future", ICOT'84, Oct. 1984 (To appear)
[Tag3] Taguchi,A.,et.al.,"The INI Internal Network and A Bird's-Eye View of New Generation Computer-Networks", ICOT Tech. Rep. TR-004?
[Tag4] Taguchi,A.,"A Study of Writing Processes in a Non-Native Language", (In Japanese), Proc. of 29th National Conference of Information Processing Society of Japan, Sept. 1984
[Tak1] Takasu,S. and Nakahara,T.,"Programming with Mathematical Thinking", IFIP'83, Sep. 1983
[Ume1]* Umesao,T.,"Techniques of Intellectual Work", (In Japanese), Iwanami Shinsho,July 1969
[Ush1]* Ushijima,K. et.al.,"An Experimental Support Tool for Polishing Japanese Articles", (In Japanese), 28th National Conference of Information Processing Society of Japan,Mar. 1984
[War1] Warnier,J.D.,et.al.,"Entrainment de la Construction des Programmes d'Informatique",Vol.1 and 2,les Editions d'Organization,1972
[Wat1] Waters,R.C.,"The Programmer's Apprentice: Knowledge Based Program Editing",IEEE Trans. Software Engineering Vol.8 No.1,Jan. 1982
[Wei1] Weinberg,G.M. and Freedman,D.P.,"Reviews, Walkthroughs, and Inspections", IEEE Tran. on S.E. Vol.SE-10 No.1, Jan. 1984
[Wer1] Wertheimer,M.,"PRODUCTIVE THINKING", Harper & Brothers Publishers, 1943
[Wil1] Wilson,W.M.,"Beyond PROLOG: Software Specification by Grammar"
[Win1] Winograd,T.,"Beyond Programming Languages",CACM Vol.22 No.7,Jul. 1979
[Wir1] Wirth,N.,"Systematic Programming; An Introduction", Prentice-Hall, 1973
[Wir1] Wirth,N.,"Program Developments by Stepwise Refinement", C.ACM Vol.14 No.4, 1971
[Wul1] Wulf,W.A.,"Trends in the Design and Implementation of Programming Language",IEEE Computer Vol.13 No.1,1980
[Yok1] Yokoi,T. "A Perspective of the Japanese FGCS Project",ICOT Tech. Rep. TR-0026,Sept. 1983
[You1] Yourdon,E.,"Structured Walkthroghs", Prentice-Hall, 1977
[Zan1] Zaniolo,C., OBJECT-ORIENTED PROGRAMMING IN PROLOG, Int. Symp. on Logic Programming (1984)
[Zav1] Zave,P.,"The Operational versus The Conventional Approach to Software Development", CACM Vol.27 No.2, Feb. 1984

(Note) *: Japanese titles are translated into English ones by the present author.

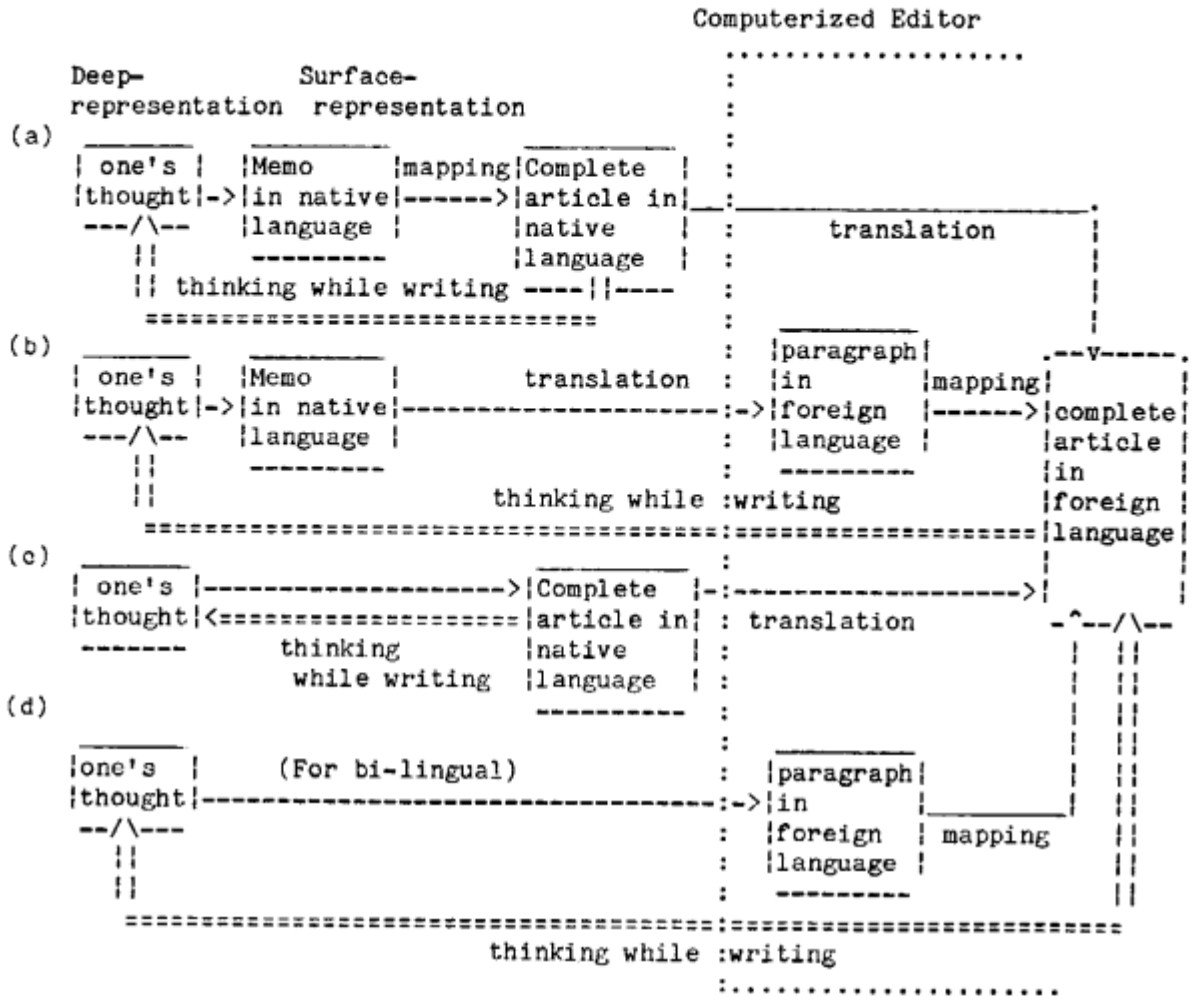


Fig.1 Processes of Writing an Article in a Foreign Language

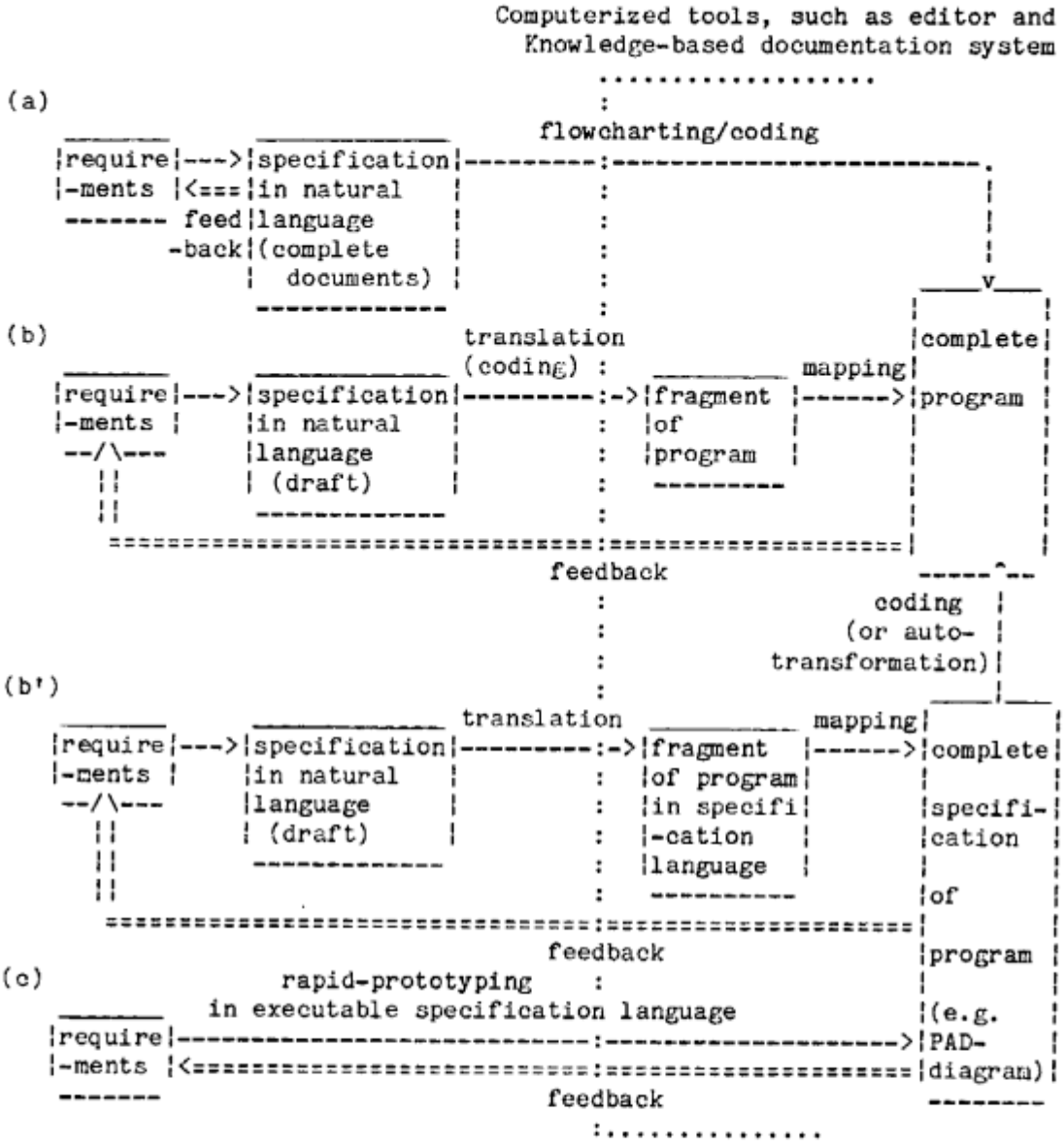
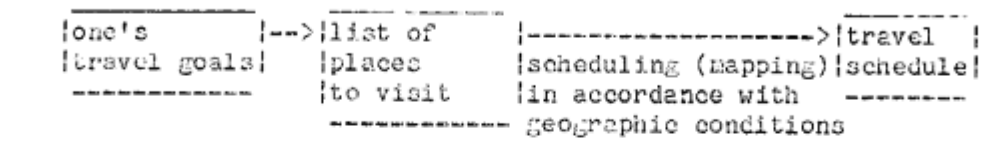


Fig.2 Programming Processes

(a) Travel Planning Process



(b) Warnier's Programming Process

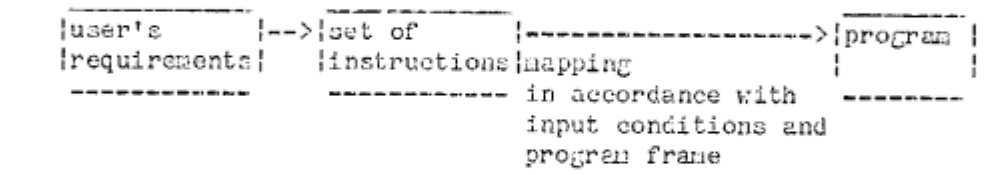


Fig.3 Travel Planning and Warnier's Programming

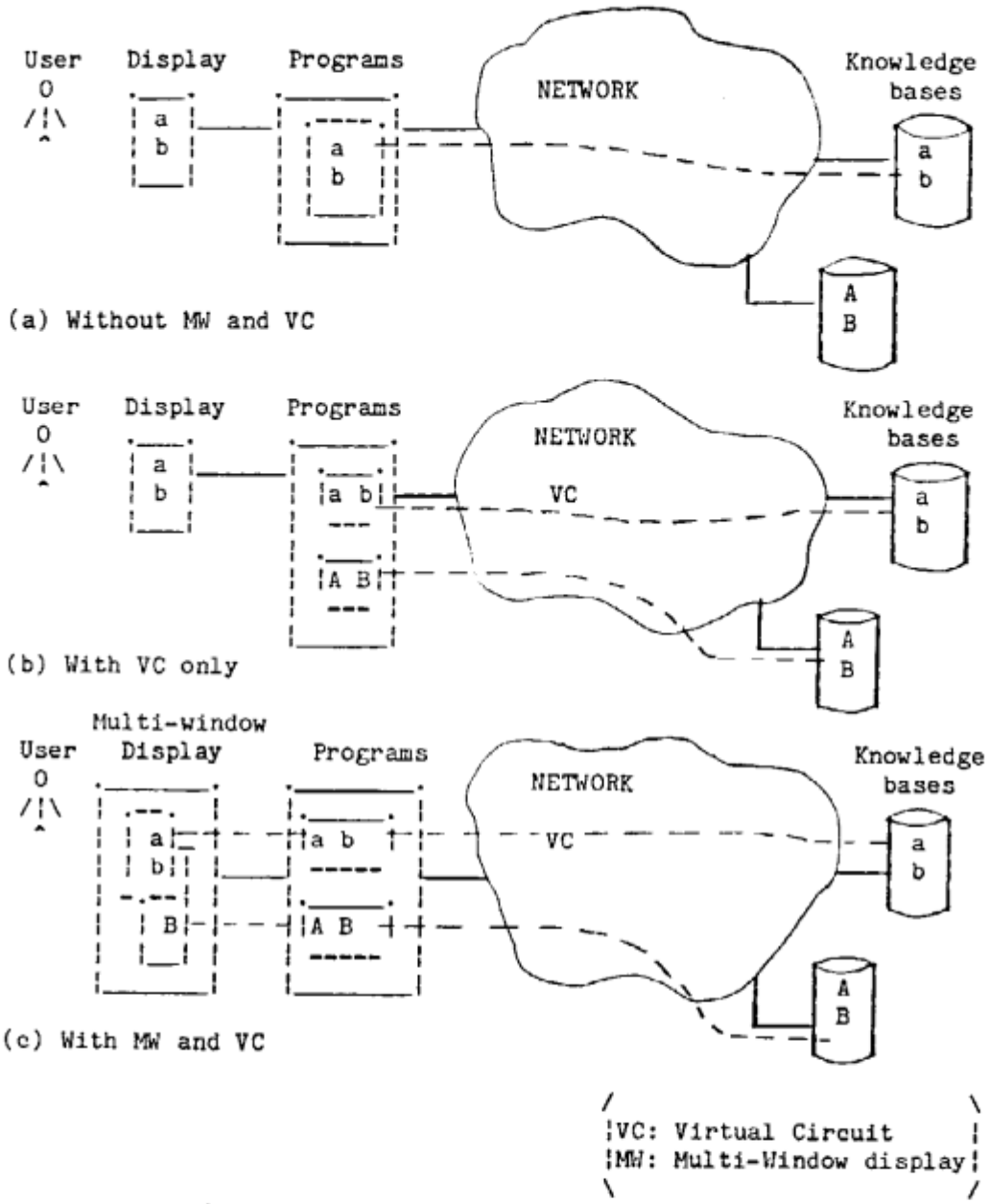


Fig.4 Visualization of Network Environment