

ソフトウェア開発過程の記録を支援する電子ノートの設計と実現

佐藤友代*, 並木美太郎*, 早川栄一**

*東京農工大学 工学部, **拓殖大学 工学部

tomoyo@namikilab.tuat.ac.jp, namiki@cc.tuat.ac.jp, hayakawa@cs.takushoku-u.ac.jp

本報告では、従来のノートのメタファを用いた、ソフトウェア開発過程における記録を支援する電子ノートシステムについて述べる。RAD や少人数グループによる開発では、明確な仕様書があるとは限らず、設計やその理由などを個人で管理する必要がある。そのようなソフトウェア開発者から、過程を記録するだけでなく、メモの検索、関連ページの参照などを行えるようなソフトウェアの要求がある。そこで我々は、(1)思考の過程を残し、(2)ソースコードの上からメモや図を書くことができ、(3)プログラムの実行結果を保存することができるような、電子ノートシステムを提案した。また、このシステムの有用性を確かめるために電子ノートを実現した。

Design and Implementation of an Electronic Notebook System for Recording Software Development Process

Tomoyo SATO*, Mitaro NAMIKI*, Eiichi HAYAKAWA**

*Tokyo University of Agriculture and Technology, **Takushoku University

This paper describes the journalizing system for software development process with notebook metaphor. In the rapid or small group based software development, developers have to manage their memorandums. They require some functions such as recording processes, retrieving the pages, and browsing related pages. Then we suppose a notebook system has the following features:

- (1) Saving processes of thinking,
- (2) Drawing memorandums and illustrations on source codes, and
- (3) Saving screen shots of a program.

We made the Electronic Notebook of a target application.

1. はじめに

ソフトウェア開発では、その過程における記録とその管理が重要である。記録には、開発中の経過や問題点の記録、そこから生まれたアイデア、印刷したプログラムに書き込むメモなどが含まれ、一つ一つのメモは次の過程に活かされる重要な要素となっているからである。Rapid Application Development(以下、RAD)や小人数による開発では、必ずしも明確な仕様書があるわけではない。しかし、設計やその理由、そこに至った過程などは保存され、何度も参照される。したがって、記録を支援するだけでなく、それらの検索や関連文書の参照などの支援が要求される。

最近では、CrossPad[1]やSmartQuill[2]などペンインタフェースを用いた入力装置がある。共に、紙に描いた内容と同じものが電子化されデータとして蓄えられる。手書きデータをテキストデータへ認識するなどの応用例はあるが、これらは主にルーチンワークに用いられるものであり、ソフトウェア開発などの創作活動を支援するようなものではない。

そこで我々は、ノート(断りがない場合は従来の紙のノートを示す)に着目し、関連ページへのリンクや検索補助の機能を有する電子ノートを提案した。また、それを開発支援の機能を持つノートとして発展させた。

本報告では、特に RAD や小人数によるソフトウェア開発での記録を支援する、電子ノートの設計と実現について述べる。

2. 既存のツールに関する考察

2.1 ノートの問題点

以前著者らは、ソフトウェア開発中の研究ノートの利用形態を調査するために、所属する研究室の教官及び学生 19 人に対してアンケートを行った。その結果、明らかになった問題点は次のとおりである。

- (1)以前に書いた内容の検索が難しい
- (2)ノート以外に書いたメモを紛失してしまう

- (3)関連のあるページをすぐに参照できない
- (4)上から書き込みすぎて思考の過程を追えない
- (5)プログラムのログなどの記録が面倒である

また、アンケートの結果にはなかったが、ノートのページには物理的な制限があると考えられる。図や表、処理の流れを書くときにその途中で切れられないようなインタフェースが求められる。

2.2 既存のツールに関する考察

次にソフトウェア開発に用いられるツールの問題について述べる。

(1)テキストエディタとノートの比較

開発中の記録をとる場合、テキストエディタを利用する方法がある。実際にメモをとるときには、キーボード入力が速く清書の必要もないという意見がある。しかし、テキストエディタで思いついたばかりのデータ構造や処理の流れ図を書くことはできない。このことから、思いついたアイデアを表現するには紙とペンのように図も字もすぐに書けるような自由度のあるものが適している。

(2)統合環境

Visual Studio[3]のような統合開発環境では、GUI プログラミングを支援するような機能を備えたエディタや、プログラムをトレースし変数の中身を参照できるデバッガなどが連携している。

しかし、GUI プログラミングにおいては変数の状態よりもむしろ画面の状態がどうなっているかを知りたいという要求がある。

このようなことから、開発の各過程で、ユーザが思考するときに必要なと判断した情報を保存し、必要に応じてそれらを参照できるようにすることを本研究の目標とする。

3. 電子ノートの設計方針

前述した考察から、次のような方針を定めた。

- (1)ペンユーザインタフェース(以下、ペン UI)を用いて思考を中断させない

電子ノートでは従来どおりの手書きが向いて

いる。思いついたアイデアや図を書き込むときに、いきなり清書するのでは思考の妨げになるからである。ただし、必要に応じてキーボードから入力することもある。

(2) 書込みの順番を時系列で管理する

既存のページは上から何度でも書き込める。このことを何度も繰り返すことにより、そのページの時系列が崩れる。ページ自体は時系列で管理されているが、その中に書かれた内容の思考の流れがわからなくなる。そこで、書込みもすべて時系列で管理する。

(3) 書き込まれた内容をすべて保存し削除しない

開発記録は個人の創造物であり、知的財産であるので、その中身が勝手に消えては困る。そこで、すべてのデータを残すことを方針とする。

(4) 後から整理することができる

数あるページを分類したいという要求がある。しかし、すべてのページが種類のどれかに属すとは限らない。そこで、まずは書込みを行えるようにして、後から整理できるようにする。

4. 電子ノート の概念

図1に電子ノートのシステム構成を示す。

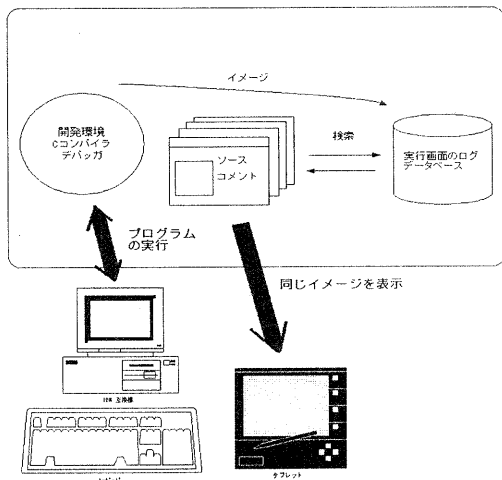


図1. 電子ノートのシステム構成

ユーザは、開発の傍らでペン UI を備えた電子ノートにより記録をとる。電子ノートは、プログラムの実行を要求し、実行画面を保存するインタ

フェースも提供する。保存された画像は日付やプログラムなどとともにデータベース化され、そこからほしい実行画面を検索し考察できる。また、電子ノートはページに対しても検索機能を持つ。

5. 本システムの特徴

前述した問題を解決するために、次のような特徴をシステムに持たせる。

(1) ページをノートのメタファとして仮想化する

書き込む作業空間であるページを従来のページのメタファとして提供する。電子化することで従来のページの物理的な制限をなくし、また紙の特徴である図や絵、文書のコピーやはりつけなどはそのまま採用する。さらに、検索やリンク機能のインタフェースを提供する。

(2) 思考の過程を追うことができる

書き込んだストロークをすべて時系列で管理するので、再生するとどのような順番で現段階に辿り着いたかがわかる。

6. 電子研究ノート の設計

設計方針に従い、次の機能を設計した。

(1) 電子ノートにおけるページ の概念

1枚のページに書く量は、内容により物理的なページウィンドウより大きくなることが考えられる。

しかし、紙のメタファではスクロールがない。紙が物理的にウィンドウの制限を越えた場合、システム側がそれよりも紙の論理領域を大きくし、スクロールすることで全体のどこを見るか決定する方法がある。書くときも同様である。

また、一つのページがウィンドウのサイズを上回っていても、物理領域の大きさで区切って順番に表示するという解決方法もある。

後者の方が紙のインタフェースには近いが、図や表、処理の流れなどを書いているときには結局途中で物理的な制約をうけることになりかねない。思考の流れを妨げない方針から、前者を採用する。

また、電子ノートでは、アイコンボタンを押し

て各機能を利用することができる(図 2 参照)。電子ノートは、検索の補助機能、関連ページへのリンク機能などを有する。さらに、ソフトウェア開発の記録を支援する機能も含む。

(2)検索の補助

計算機における機能の一つに検索機能があるが、一般的に検索処理はテキストに対して行う方が便利である。また、大量のデータに対して行うときに効果を発揮する。

他方、電子ノートをノートのメタファとして扱う場合、絵やデータ構造図、イメージなどの非定型なデータを扱うので、検索の問題が解決しないように思われる。

そこで電子ノートでは、検索の補助として、時間の指定やそのときに書き込んだ内容の有無をに着目した。これにより参照したいページを絞りこむ。

次に、検索条件として何を表示すべきか述べる。例えば、時間指定ならば「○月△日×時」や「昨日の午前中」などさまざまな表現がある。このとき明示的に日付と時間を指定するのもよいが、過去を振り返るときには「～前」とよぶ後者の方がより日常の表現に近い。

また、内容の種類の有無をチェックするインタフェースを提供する。内容の種類とは、ページの中心にソースプログラムを含んでいるか、ほかのページのコピーを含んでいるかなどである。これらはユーザの記憶を便りにチェックしてペンで操作してもらうものである。特に何も指定しないときはその条件を無視する。

図 2 を見てもわかるとおり、条件のところをペンで指定するだけで、簡単に的を絞ることができる。ただし、テキストの検索にはキーワードを与える方が確実なので、キーボード入力で指定することにする。

検索結果は、全体の見通しのよいように、サムネイル表示を行う。

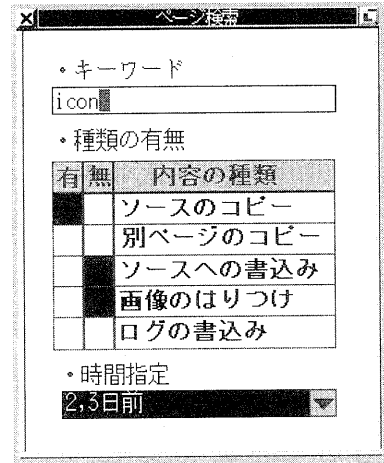


図 2. 検索の条件設定画面

(3)プログラムの実行結果の取込み

本機能では、プログラムのテストを行い、実行画面のイメージを取り込む。また、取り込んだ画像をページにはりつけられるようにする。これにより、デバッグ中にほしい情報を集めるほうに集中し、その後で、バグに対する考察に集中することができる。

この機能と(4)のリンク機能を組み合わせることにより、実行結果やそのソースに関するページなどの関連づけを行い、互いのページをすぐに参照することも可能である。

プログラムの実行画面を取り込む手順は次のとおりである。

- ①プログラムの実行ボタンを押し、選択画面から実行プログラムを選択する。
- ②画面取込みボタンを押し、ほしい画面の状態を保存する
- ③結果表示用ウィンドウに時間順に表示される

プログラムによっては引数を必要とするものもあるので、その場合はユーザにキーボードを使って入力してもらう。2 回目以降はシステムが自動的にそのプログラムを起動する。これにより、実行プログラムの選択時間を省略することが可能である。

また、プログラムの実行画面に動きがあるものも考えられる。そのような場合、処理は重くなるが画面がフラッシュされるたびにその画面の状

態を保存することで対処できる。

保存した画面は、実行コマンドや時間とともにデータベースとして保存される。ユーザは保存した画像の必要な部分だけをはりつけるなどしてバグに関する考察を行う。また、保存した画像を見て以前に発見したバグに似ているものが会った場合、データベースを検索し、そこからバグの原因を逆参照するというような使い方も想定している。

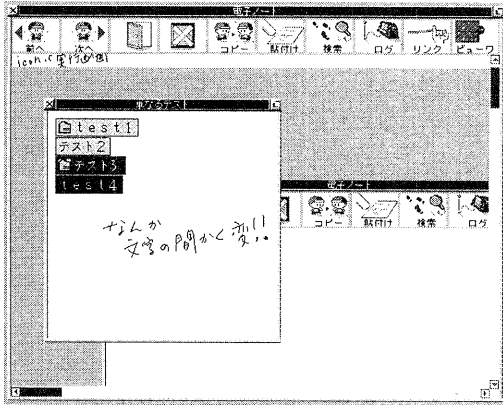


図3. プログラムの実行結果のはりつけ

(4) リンク機能

検索とは別に、関連する箇所をすぐに見たいという要求がある。そこで、ユーザが指定した関連のあるページを相互に参照しあえるようなリンク機能を提供する。

これは結果的に、一度リンクを設定した箇所は検索しなくてもすぐに参照できるという意味でも利点大きい。

リンクの設定方法について図4に示す。

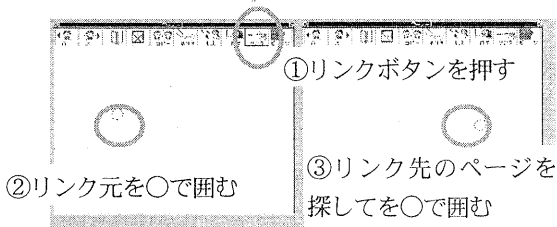


図4. リンクの設定手順

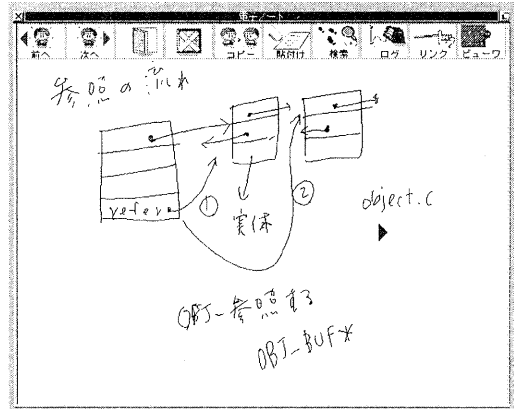


図5. リンク情報の見え方

しかし、これではどのようなページにリンクするのかわからない。そこで、リンクボタンの少し下に、その情報としてリンク先のページのリンク位置付近の内容を表示する。これにより、ユーザがリンクを指定するときどこにリンクするのか、その説明を考える必要がなくなる。

リンクをたどるときは、リンクボタン(図5)の近辺にペンを近づけるとリンク先の情報を数秒間表示するようにする。ペンで押すと、関連ページを表示する。

(5) ソースプログラムへの書き込み機能

サンプルやデバッグ中のプログラムなどの、データ処理の流れを図を描いて確かめることはよくある。デバッガなどで変数の中身を確認して動作が怪しいとわかったときでも、自分なりに処理の流れ図を描いて、訂正することがある。

しかし、ソースの上からそのまま書き込むことはできないから、ソースの上にページを重ねて表示する方法を考えた。しかしこの方法では、ソースが変更されると実際に上から書き込んだデータとの対応関係がずれてしまう。したがって、ソースの上から書き込む場合はソースの内容をコピーしてから書き込む必要がある。図6にソースプログラムへの書き込みの例を示す。

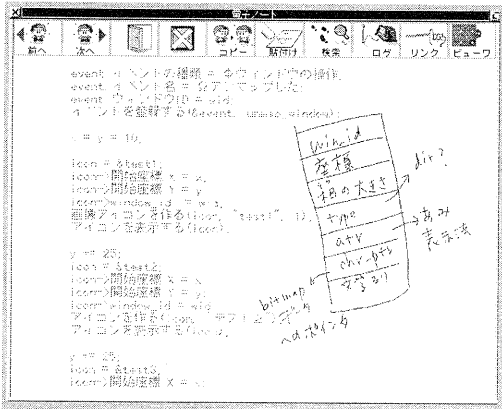


図6. ソースファイルへの書き込み

7. 実現

特に重要であると思われる電子ノートのリンク機能、プログラムへの書き込み機能、ソフトウェアの作成からデバッグ、テストの過程で思考過程を追えるようにした。

電子ノートが提供するページの各機能の有効性を確認するために、ペン UI をサポートするオリジナル OS、OS/omicron 第4版(以下、V4)[4]上で電子ノートを作成した。

- ・リンク機能
リンクの設定は、リンクボタンを押して関連づけたい2箇所を○で囲むだけでできるようになっている。リンク先の情報に関しては、今のところユーザには意識する必要のないページの名前が表示される。今後、設計したとおり、リンク先付近の内容を表示するように変更する必要がある。
- ・ソースへの書き込み機能
ソースを書き込む際に、プログラムにページを重ねて表示させる方法とそのプログラムの内容をページにコピーする方法が考えられていた。今回はソースが変更されるとソースとページの部分部分の対応がとれなくなり一貫性が崩れてしまうので、コピーして書き込むことにした。
- ・思考の過程の追跡
思考の過程を追えるように、過去のページに書き込みを行う場合は新しいページにコピーして

その上から書き込むように実装した。リンク機能を用いることでこのような思考の流れを持つものの経過を追えなくはないが、実現したリンクは1対1の相互リンクであるので、思考の流れや論文の章立てのように一連の流れを持つようなリンクに適さないかもしれない。それよりは、ページにカテゴリという属性を与えて、同じ属性のページの流れを表現するリンクとして実現することも記録の整理に役立つと考えた。

8. おわりに

本報告では、ソフトウェア開発中の開発過程の記録を支援する電子研究ノート的设计とその実現について述べた。

今後、今回出てきた問題に合わせてシステムに変更を加え、実現した機能とそのユーザインタフェースに関する定量的な評価、グループによる開発支援を行う予定である。

[参考文献]

[1]<http://www.ikeshop.co.jp/plaza/and/crosspad/index.html>
 [2]<http://www.innovate.bt.com/showcase/smartquill/>
 [3] Visual Studio 6.0, Microsoft, 1998
 [4] 早川栄一他: “手書きインタフェースを支援する OS/omicron 第4版の構成”, 情報処理学会第4回コンピュータシステムシンポジウム論文集, pp. 35-42, 1992
 [5] 佐藤友代他: “ソフトウェア開発を支援する電子研究ノートの実現”, 第58回平成11年前期全国大会, 1999
 [6] 二村尚他: “手書きユーザインタフェースの評価を支援する環境の実現”, 情報処理学会ヒューマンインタフェース研究会 61-4, 1995
 [7] 野口悠紀雄: “「超」整理法”, 中公新書, 1993
 [8] 中島一彰他: “ペンインタフェースを用いた分散 KJ 法システム”, プログラミングシンポジウム, 1995