よりよいコメント記述のためのプログラミング環境
―文書からコメントを分離することによる効果―

1．背景

プログラムにおけるコメントは、そのプログラムの挙動を説明するために記述されるものである（一時的に式・文を取り除く用途でのコメントはコメントアウトとして、本文書中では区別する）。よいコメントを記述しておけば、後でそのプログラムを読むときに大変有効であり、自分も含めそのプログラムを読む人のことを考えると、欠かせない作業である。

ここで、現状のコメント記述について考えてみると、プログラミング言語で定義された記号を用いて、テキストでプログラム本体と入り交じって記述することとなっている。たとえば、C言語や Java で記述されたプログラムであれば、/* と */ で囲まれたテキストをプログラム中に記述することで、コメントを記述する。このようなコメントの記述法には、以下のような問題がある。

- テキストでの記述に限定されることによる問題
  図示すれば明確に説明することができるプログラムであっても現状ではテキストで記述しなくてはならない。コメント記述の表現力が低いことは、コメント記述が人間に対する記述であるということを考えると、大きな問題である。
- プログラム中に入り交じって記述することによる問題
  あるプログラムに対してコメント記述も一つに限られてしまう。たとえば，あるプログラムに対して、日本語用のコメントや英語用のコメントといったように参照したいコメントが異なる場合でもそれらを切り替えることなどできない。特に日本語などの場合には記述に用いる文字コードなども影響するため、コンパイラの動作にも関係する場合もあり、深刻な問題となる場合もある。

これら二つの問題以外にも、プログラム本体とコメント記述の対応づけのために、余計な改行などを挿入し、プログラム全体のつながりなどがみにくくなってしまうなど、現状のコメント記述法には問題が多い。

2．目的

現状のコメント記述法の問題点をふまえ、本プロジェクトでは、プログラム本体からコメントの記述を分離して保持することにより問題の解決を目指す。

コメントの分離により、コメントの記述がテキストに限定されないため、図示によるコメントが可能となる。また、あるプログラム本体に対して、複数のコメントを切り替えて表示することもできるようになる。

本文書では、提案機構をプログラミングでのコメントに適用した場合を中心に説明を行うが、実際には本プロジェクトで提案する機構の適用範囲はプログラムに対するコメントの記述のみでなく、任意の文書に対するコメントの記述へと適用できる。

3. 開発の内容

分離したコメントをウィンドウシステム上で適切に対応づけを行い表示し、編集できるプログラミング環境の実現した。そのために、分離されたコメントを表現するためのファイル形式の策定も必要であった。

本プログラミング環境の基本となるのは、プログラムを表示しているエディタに対する編集やスクロールなどの操作により、対応するコメントが表示されるように調整する機能である。また、コメントの貼られているプログラムが一目で分かる機能，コメントの表示・非表示を切り替える（アイコン化）機能、コメントと対応するプログラム間での移動など、コメントの記述を補助する機能がある。また、コメントの編集に関しては、プログラム本体の編集操作と統一的にできることを目指して設計されている。特にコメントの図示に関しては、簡単な図形エディタを作成し、手軽な操作で描くことができるように実現した。

上記の設計のもと、既存のエディタである Emacs を拡張する機能を利用することで提案プログラミング環境を実現した。Emacs Lisp という拡張用言語を用いて、Emacs が提供している機能のほとんどの部分を拡張することが可能である。また、コメントの図示などグラフィカルな表示を必要とする一部の機能に関しては、Java Swing によるプログラムと Emacs を協調させることにより実現した。

本プロジェクトで開発したプログラミング環境は、プログラムを表示するエディタ（以降プログラムエディタ）とコメントを表示・編集するためのフレーム群（以後コメントフレーム）からなる。図 1 に開発したプログラミング環境の動作例を示す。

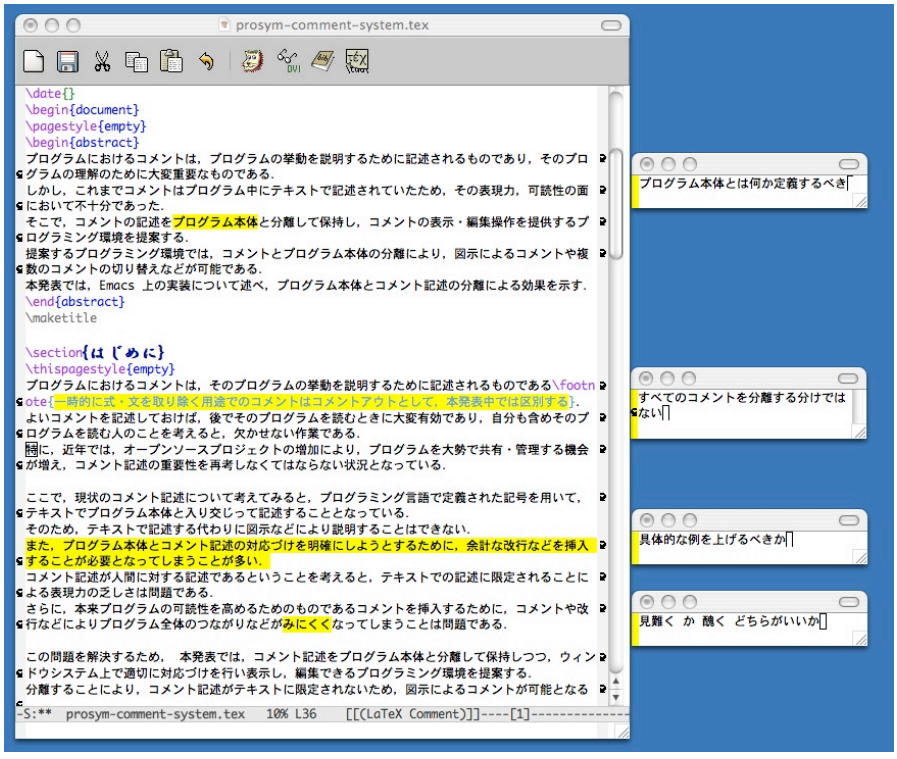

図 1: 動作例

左側のフレームがプログラムエディタで、右側に並んでいるのがコメントフレームである。この例では、表示中のプログラム本体中の四ヶ所にコメントが張られている。また、左側のフレームでは、コメントの張られた領域は背景色により、コメントが張られていることが分かる。さらに、右側に並んでいるフレームは対応するプログラムの横に表示されるようになっていることからもコメントの張られているプログラムとの対応づけが可能である。

本プログラミング環境の全体のシステム構成は図 2 のようになる。

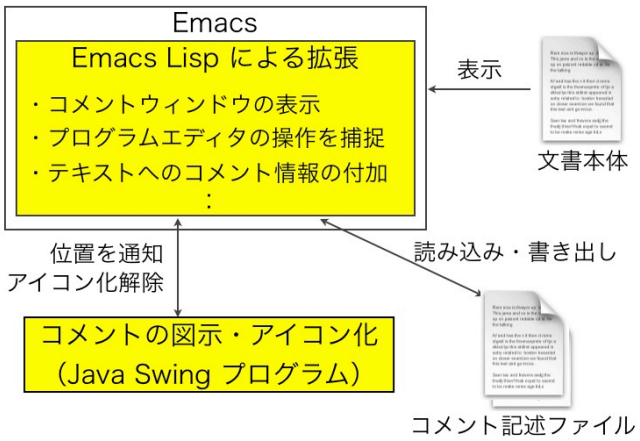

図 2: システム構成

まず、分離されたコメントは、複数のコメントをまとめて一つのコメント記述ファイルとして保存する。たとえば、コメント記述ファイルは日本語コメント記述ファイル、英語コメント記述ファイルのように用途によってまとめ、それぞれを選択して表示可能である。

本プログラミング環境の基本となるのは、プログラムを表示しているエディタに対する編集やスクロールなどの操作により、対応するコメントが表示されるように調整する機能である。プログラムエディタには既存のエディタ Emacs を採用し、その拡張機能を用いてエディタ上の操作を捕捉する手法により、プログラムとコメントとの対応づけを行う。

また、各コメントフレーム上での編集にも Emacs を用いる。このことにより、プログラムの編集とコメントの編集を統一的な操作（キーバインディングなど）で行うことができ、ユーザビリティを高めている。また、本開発成果の特徴である図示によるコメントについては、Emacs のみでは実現できないため、Java Swing によるプログラムを協調させることにより実現した。

このように、Emacs と Java Swing のみで実現することで、Mac OS,Windows , Linux など幅広いシステム上で動作することが期待できる。

4. 従来の技術（または機能）との相違

コメントの記述を考慮した既存の製品としては、Microsoft Word や Apple Pages などのワープロソフトが上げられる。これらの製品では、文章に関連付けてコメントを付けることが可能ではあるが、本プロジェクトで提案したようなコメントの図示や文書本体との分離については考えられていない。そのため、上記で述べたような利点を持ってはいない。

5. 期待される効果

これまでプログラミング言語の一部として実現されていたコメントをプログラムから分離し、プログラミング環境により補助することで対応づけができるようになったことが本開発の成果である。

まず、コメントを分離することにより、以下のような利点がある。

① 読み手に合わせてコメントを取捨選択ができるようになった。例えば、日本語のコメントと英語のコメント、初心者向けのコメントと上級者向けのコメントなどを別々に用意し、読み手が自由に選択できる。

② コメントの配布を文書とは別にすることができる。たとえば、部外秘のコメントを作成した場合でも、文書本体と分離されているので、文書本体のみを公開することが可能となり、コメントを外部に出さないで済む。

③ プログラムの記述者とコメントの記述者が分業できる。ただし、プログラムとコメントの対応づけを取る必要上、改変されたプログラムとコメントの対応づけについては今後の課題である。また、コメントが図示できるようになったために、より分かりやすいコメントを記述可能となった。

6. 普及（または活用）の見通し

Emacs は広く使われているエディタであるので，実現した拡張を組み込んでもらうことで多くのユーザに利用されることが期待できる。現状では、実装上の不十分な点を改善しているため公開に至っていないが、なるべく早く公開できるよう作業をすすめている。

また、コメントファイルの仕様を公開することも予定している。その仕様にあうようにして別のエディタや統合開発環境へとコメントを読みこむ機能を追加できれば、ユーザはさらに増えると予想できる。

7. 開発者名（所属）

高野 保真（電気通信大学 大学院）