

エディタにおける、利用者モデル構築の一手法

1B-7

小高知宏 加藤芳隆 内山明彦

早稲田大学理工学部

1. はじめに

我々は従来よりソフトウェア自動生成系の開発を目標として、ソフトウェア開発環境について研究を行ってきた。これまでに、要求仕様定義段階からテスト・デバッグ段階まで種々の段階において開発環境に対する提案を行ってきたが、今回は、マン・マシンインタフェースの中心であるエディタシステムについて提案を行なう。エディタはソフトウェア開発のすべての段階において利用され、他の開発ツールとユーザとの接点となるものである。したがってエディタの優劣を考える場合には、編集能力そのものが優れている事も重要であるが、いわゆる「使いやすさ」も評価の重要な指標となる。

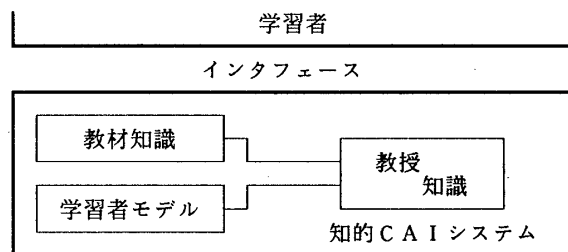
こうした観点から、ユーザの熟練度やユーザのエディタに対する知識などを推定し、この結果を利用者モデルとして評価し、これをもとに編集操作の支援を行っていくエディタの構想について既に報告した^[1]。今回はこのエディタに関して、特に利用者モデル構築の手法に関して報告し、本手法を用いた場合にエディタの使いやすさがどのように向上するかを考察した。

2. 利用者モデル構築の手法

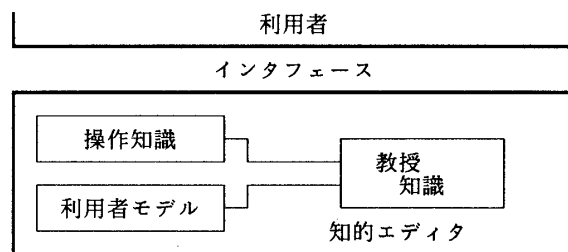
本エディタが利用者モデルを構築する手法は、知的C A Iシステムにおける学習者モデルの推定手法と一致する点が多いので両者の比較により構築手法を説明する。

一般に知的C A Iシステムで用いる知識は図1(a)に示すように教授知識、教材知識、学習者モデルから構成される^[2]。これはそれぞれ学習指導に関する知識、教えようとする学習内容に関する知識、学習者の理解程度を推定した結果に対応する。これを今回報告するエディタが用いる知識に対応させれば図1(a)と図1(b)との比較で示したように、教材知識はエディタの操作に関する知識、教授知識は利用者に対する支援に関する知識、学習者モデルは利用者モデルに対応する。

知的C A Iの分野では学習者モデルとして多くのモデルが提案されているが、今回のエディタではそのうちオーバーレイモデルとパータベーションモデルをエディタ向きに変更したものを採用した。オーバーレイモデルは学習内容に対する理解状態を単純に評価するものであるが、エディタにおいては図2の「コマンド利用頻度解析」に対応する。すなわち単純に各コマンドの利用回数を数え挙げ、[1]で述べたようなヘルプ機能への応用等に利用する。知的C A Iにおけるパータベーションモデルに対応するのは、コマンドの履歴を解析し、特定のパターンを検出する「コマンド列解析」に対応する(図2)。これは取り消しキーが押される直前の入力や、利用頻度の高いコマンド列に対して、新たなキー定義を与える際に利用する。これによりエディタが新しいコマンドを学習して高機能にな



(a) 一般的な知的C A Iシステムの構成



(b) 利用者モデルを考慮したエディタの構成

図1 知的C A Iシステムと、利用者モデルを考慮した知的エディタ

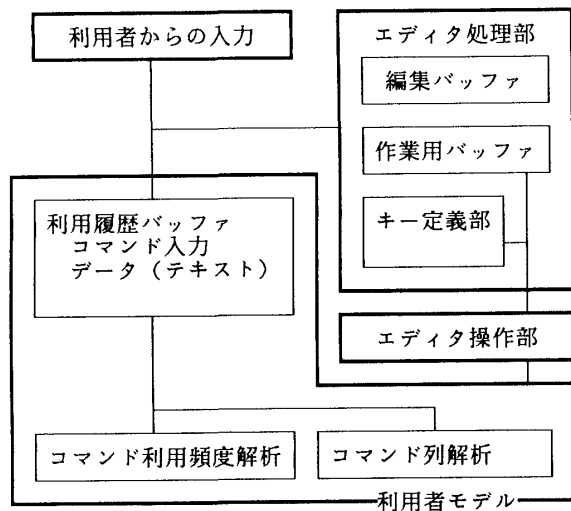


図2 利用者モデルと知的エディタシステムの構成

り、また本来備えている編集機能に対するキ一定義を、ユーザの期待するものに変更することが可能である。

実際のモデル構築過程では、コマンド入力や入力データ等ユーザからのキー入力をすべて保存する「利用履歴」バッファを用意し、これを適当なタイミングで解析することでコマンド利用頻度解析とコマンド列解析を行い、ユーザモデルを構築する。

3. 本手法の利点

文献 [3] ではマン・マシンインタフェースにおける一般的問題点を考察しているが、これにそって本エディタの利点を考察する。表1に一般の問題点と、これに対応するエディタにおける問題点を示す。この内①～④に

関しては本エディタのヘルプ機能で対処できる。本エディタでは、コマンド利用頻度解析を行なうことで適切な量のメッセージを適切なタイミングで利用者に提示できるため、表1のような問題は生じない。⑤については、コマンド列解析に基づくエディタの変更で対応出来る。本方式ではエディタが自動的に利用者の潜在的な要求を検出して、利用者に提案する形でインタフェースを変更する。すなわち、利用者が具体的に変更を指示する必要がないため、エディタのように利用者の層が幅広い場合特に有効である。

4. おわりに

本エディタは現在Unix環境上でemacsを利用して構築中である。今後実際の運用を通して有効性を確認する予定である。

参考文献

[1] 粟根、小高、内山：“利用者モデルを取り入れたエディタの考察”、情報処理学会第37回全国大会、4H-5、pp.1300-1301。
 [2] 溝口、角所：“知的CAIにおける学習者モデル”、情報処理、Vol.29, No.11, pp.1275-1282。
 [3] Gerhard Fischer：“Human-Computer Interaction Software:Lessons Learned,Challenges Ahead”,IEEE Software,Vol.6 No.1, pp.44-52(January 1989)。

番号	高機能システムにおける問題点	エディタにおける問題点
①	利用者がツールの存在を知らない	利用者がコマンドの存在を知らない
②	利用者がツールの呼び出し方法を知らない	利用者がコマンドの呼び出し方法を知らない
③	利用者がツールの利用方法を知らない	利用者がコマンドの利用方法を知らない
④	利用者がツールの出力を理解出来ない	利用者がコマンドの結果を理解できない
⑤	利用者はツールの結合や変更が出来ない	利用者はコマンドの結合や変更が出来ない

表1 高機能システムにおける一般的な問題点と、エディタでの比較