

ソフトウェア品質評価システムESQUT

— 知識工学を応用したソフトウェア保守性の評価 —

6E-3

佐藤 誠 山田 淳 春木 和仁  
 ㈱東芝 青梅工場 ㈱東芝 システム・ソフトウェア技術推進部

1. はじめに

現在ソフトウェア品質の諸要素のうちで、現場で定量的に管理されている要素として、信頼性と性能がある。しかし、これら以外、例えば、プログラムの使い易さ、保守し易さといったような品質の諸要素についてはまだ評価方式が定式化されていない。また、何らかのモデルを導入して定式化したところで、出力された数字を見ても、それを開発、保守にどのように役立ててよいか分らないという問題があった。

このような定式化が困難な品質の要素を扱う場合、現場でソフトウェア開発を行っている専門家の知識を利用した評価の方がむしろ妥当であると考えられる。また、単なる定量化のみでは十分な効果が期待できないような場合は、むしろ注意点の指摘や品質向上のための対策の指示が重要となってくる。以上の理由からエキスパート・システムの導入は、この場合の有効な解決策となり得る。

ソフトウェア品質評価システムESQUT<sup>[1]</sup> (Evaluation of Software Quality from User's viewpoint)のサブシステムESQUT-II<sup>[2]</sup>は、プログラムのソースコードを静的に解析し、ソフトウェアの保守し易さに関係した指標の定量化を行うが、今回、このESQUT-IIを用いてソフトウェアの保守性の診断・評価を行うエキスパート・システムを開発して、その実用化の可能性を探ることにした。

2. 評価システム

本評価システムの構成は、定量化ツール部とエキスパート・システム部からなる(図1参照)。定量化ツール部にはESQUT-IIを用いる。エキスパート・システム部は、診断型エキスパート・システム構築ツールPROBE<sup>[3]</sup> (PROlog Based Expert system tool)により構築した。PROBEは専用の言語で記述した知識からplorog記述の知識ベースと、ユーザとのインタフェースを司るメニューサブシステムを作成する。知識表現は、if-then ルールを基本にしており、知識の記述言語は「あいまいさ」を自然に記述できるよう設計されている。

評価の手順は、まずソースコードを定量化ツールESQUT-IIによって解析し、定量化した指標を得る。次にこの定量化値をエキスパート・システムへの入力として評価および品質向上施策に関する推論を行う。定量化ツール部は既に開発済みであり、今回はエキスパート・システム部を新規に構築した。

3. 本エキスパート・システムの特徴

エキスパート・システムの導入の動機については前述したが、ここでは今回PROBEを用いて構築したエキスパート・システムの特徴について説明する。従来用いてきた評価方法と大きく異なるのは以下の三点である。

・対策の指示

従来方式では品質指標に関して定量化するのみであったのに対し、本エキスパート・システムでは不具合状況の予測、対策の指示を行う。

・if-then ルールによる評価

複数指標から評価を行う場合、線型合成する方法がよく用いられるが、ソフトウェアにはいろいろなケースがあり、各指標の値のバラツキも激しく、妥当な評価が得にくい。そこで、専門家の知識を利用する訳だが、このための知識表現にはif-then ルールを用いている。

・ファジィ導入

知識を表現する場合あいまいさが含まれることが多い。例えば、「if モジュール数が多いと then 管理上の不具合が起こる」というルールを考えた場合、条件部の「多い」という言葉は何個以上なのかあいまいである。また、モジュール数が多くても、管理上の不具合が起こらない可能性があれば、このルール自体にあいまいさがあるということになる。これらのあいまいさを表現するためにファジィを導入している。

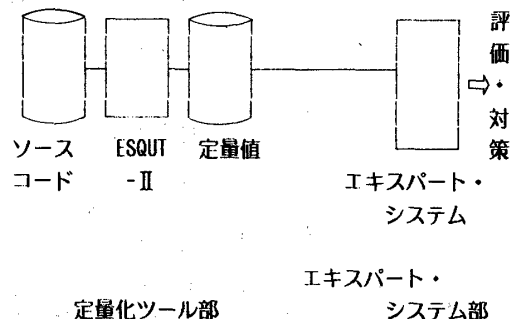


図1 評価システム

"ESQUT" Software Quality Evaluation System - Knowledge Engineering Approach to Software Maintainability Evaluation -

Makoto Sato, Atsushi Yamada, Kazuhito Haruki  
 TOSHIBA Corp.

4. 評価の対象

評価の対象は、C言語で記述したソフトウェアのソースコード自体であり、以下の三つの観点で評価を行う。

- ・プログラムの物理的構成  
プログラムのファイル（コンパイル単位）分割のしかたについて評価する。
- ・プログラムの論理的構成  
プログラムを構成するモジュール（C言語で言うところの関数）間の結びつきについて評価する。
- ・構成単位自体の内容  
プログラムを構成するモジュール自体について評価する。

これら三つの観点から知識の収集を行った。

5. 推論方式

PROBEでは、多段階の推論ができるエキスパート・システムを構築する。今回作成した例では、先ず第一推論で保守時に発生すると思われる障害の診断を後向き推論で行い問題点を明らかにする。次に第二推論で直前に推論された結果から前向きに推論を行うことによって障害を回避するための対策を得る（図2参照）。

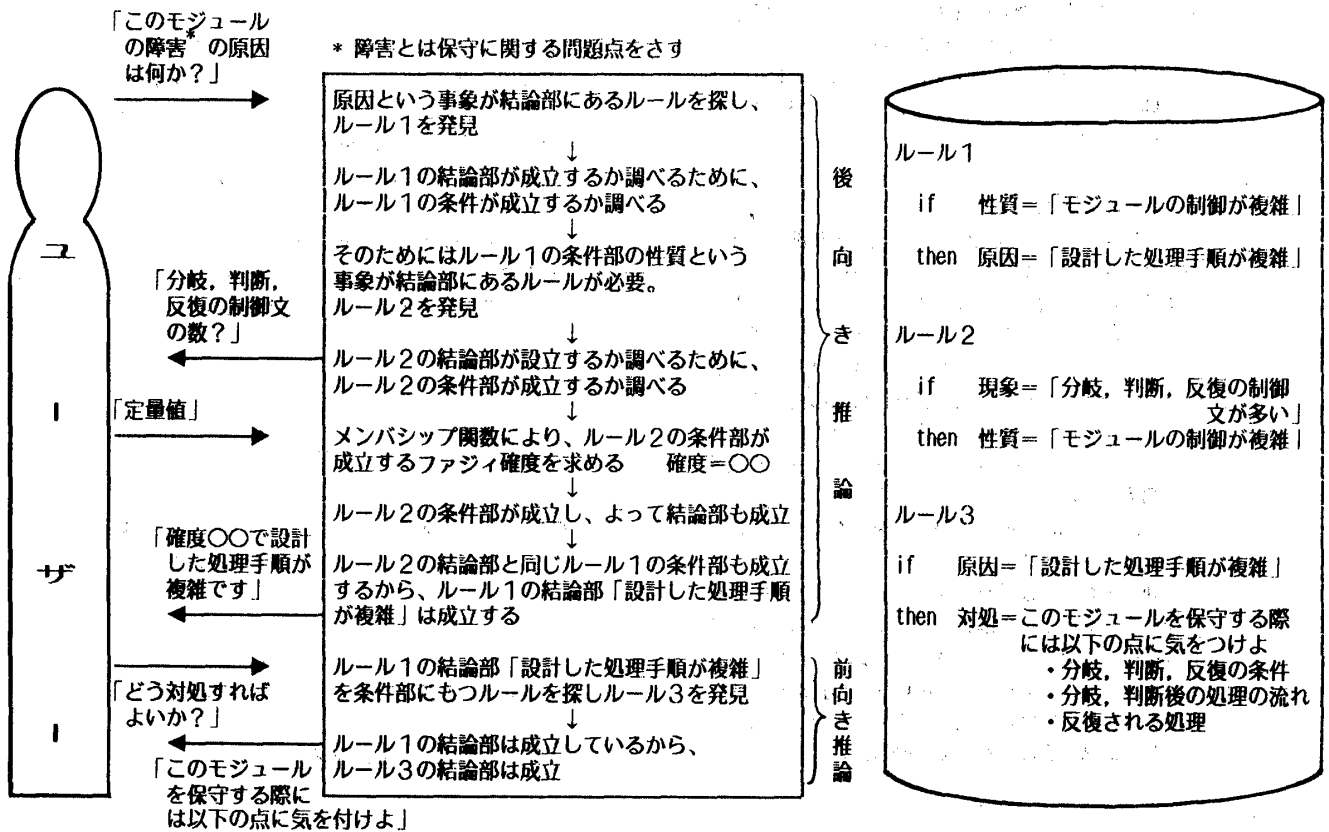
6. おわりに

今回ソースコードを静的に解析した結果を用いてプログラムの保守性を診断するエキスパート・システムを構築した。これはあくまで基本的な評価ルールだけからなるものであり、新規にルールを追加することによって拡張していくことが可能である。

今後の課題は、このような評価ルールを追加、改良し、各開発部門で実用化していくことである。これには品質評価のための普遍的なルールを見出すと共に、使用する部門の専門家から、その部分の使用者毎の独特な評価ルールを抽出するための方法論を確立していく必要がある。

参考文献

- [1] 中村 他,  
“ソフトウェア品質評価システムESQUT(1)  
—品質評価モデルデータフロー構築—”,  
情報処理学会第29回全国大会5R-2, 1984
- [2] 山田 他,  
“ソフトウェア品質評価システムESQUT  
—保守し易さの評価法—”,  
情報処理学会第30回全国大会7S-3, 1985
- [3] 小島 他,  
“エキスパート・システム構築ツールPROBE(1)  
—全体構成—”,  
情報処理学会第32回全国大会5L-1, 1986



エキスパート・システム

ルールベース

図2 推論のしくみ