

ソフトウェアデバッグエキスパートシステムにおける
知識表現
4G-6

土田賢省 今井恵一 赤尾由香利セシリア

日本電気(株)

1. はじめに

デバッグエキスパートシステム(DBES)は交換系ソフトウェアの試験工程における障害やエラーに対して診断、デバッグをユーザと対話的に進めていくシステムである。我々は、DBESの開発を通じ、DBESで扱う交換系ソフトウェアのデバッグの知識に関する知識表現モデルを専門家との共同のもとに構築した。このモデルは、専門家の知識の整理方法及び整理された知識に対する利用法を示すものである。さらに、ES開発者と専門家との会話において共通の言葉の役割も果たす。以下では、この知識表現モデルについて説明する。

2. 知識表現モデル

2. 1. symptom-cause model

診断型エキスパートシステムでは、多くが事象-原因の形式に基づくルールベースにより実現されている(例、MYCIN[1])。交換系ソフトウェアにおけるデバッグの知識も事象-原因の集合(symptom-cause sets)と見なすことができる。

(1)構造 各事象にはその原因と考えられる集合が対応する。ここで事象と原因という言い方は、相対的なものである。すなわち、ある事象の原因となっているものも、それを引き起こす側からみれば事象となる。例えば、ある異常な症状が発生したとき、いくつかの原因が考えられる。それらの原因には、各々互いに区別しうる症状(テスト・結果)が対応する。さらに、それらの各症状に対しても、いくつかの原因が存在するというように階層構造を順次たどっていくことができる。また、最終的に到達した(発見された)バグに対して、修正法などを説明するものがあれば、それもそのバグの原因と見なす。

図1. a, bにsymptom-cause modelの様子を示す。Sは事象であり、c1, c2, ..., cm, c11, c12, ..., c1n, cpl, ..., cpq, c111, ..., c11r, ...は直接あるいは間接的にSの原因と考えられるものである。c1に着目すると、c1はSの原因の一つである。一方、c11, c12, ..., c1nから見れば、これらを原因の可能性の集合としてもつ事象と考えられる。

(2)結論の種類 ある事象に対する結論には2種類ある。一つをpossible conclusions、もう一つをdefinitive conclusionsと呼ぶ。definitive conclusionsは与えられた事象と1対1に対応する結論であり、それ以外の原因は考えられない。possible conclusionsは、他の原因も考えられる場合であり、それがその事象の原因であることを検証する必要がある。

2. 2. symptom-cause modelの拡張

交換系ソフトウェアにおけるデバッグは非常に複雑であり、上記の概念だけでは、この作業を表現することは不可能である。例えば、ある事象に対して、ある程度分析が進んだ後に、現在分析している部分は無関係であると結論されることがある。このとき、担当者はバグを発見するために他の部分を探し始めなければならない。このような状況に対応するために、上記モデルにnegative conclusionsの概念を加えることにした。negative conclusionsは、分析の対象を他の可能性のある原因に移すことを表す。図. 1 bにおいて、黒塗りで表現された要素がnegative conclusionsである。

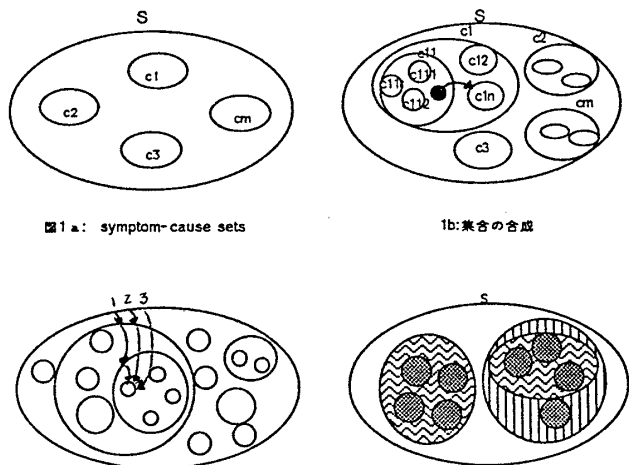


図1 a: symptom-cause sets

1b: 集合の合成

1c: 推論は外側から内側の集合に向かって進む。アークは推論を数ステップとばすことを表す。1,2,3等の推論が可能である。
1d: 事象Sに対して、黒塗りは即座に想起される原因である。まず、黒塗り集合にグループ化し、そして黒塗り集合に内側から外側の集合へとまとめられる。

3. 知識表現モデルの活用

3. 1. 推論

前記モデルに基づきどのように推論されるかを説明する。バグの現象が与えられると、まず担当者はその現象に符合する事象を最も外側に位置する集合の中から見つける。その事象が見つければ、その集合(その事象に対応)の中にバグの原因があり、バグ修正の方法も含めた最終結論もその集合の要素か、要素(集合)の要素の中にあると考えられる。このことは、推論中にnegative conclusionsが現れない限り成立する。さらに続けて、次の段階では、事象集合の要素の中からその事象の原因となるものを見つける。すると、今度は、その原因が新しい事象となるというように、推論はある結論に達するまで、外側の集合から内側の集合へと進んでいく。

担当者が交換ソフトのデバッグに慣れてくると、ある事象が与えられたとき、特定の状況下では、必ずしも一番外側の集合から1ステップずつ順に中へと進む必要がなくなることが起こる。すなわち、ある事象から何階層か中の要素へと直接進むことが可能となってくる。これは、ある状況のもとにおいて外側の集合から何段階か飛び越えて直接内側の集合へアークが張られたと考える(図. 1c参照)。初心者 비해、経験者の方がより頻繁にこれらのアークを使う傾向があると捉えることができる。これは、ユーザの習熟度の差により生じる知識の扱い(推論)の違いをモデルにより説明するものである。

3. 2. 知識獲得

(1)知識の整理

専門家がデバッグの知識をどのように(頭の中で)整理しているかについて述べる。

まず最初に事象が与えられ、その事象の原因を想起する。これらの原因は他の原因の事象であったり、または結論であったりする。専門家は原因をその特徴からグループ化し、内側の集合から外側の集合へと各要素をまとめていきそれぞれに共通の事象(名)を割り当てる。これを繰り返し行う。あるいは、逆に、一つの原因を取り上げ、それを事象として、外側から内側へと分解していく。ここで重要なのは、専門家は頭の中に初めから明確に階層的に分類された知識を持っているのではないということである。一つの事象は、即座に専門家の頭の中にある他のいろいろな事象を想起させるが、これらの事象間に必ずしも最初から明確な関係がある必要はない。個々の事象に対して、外側の集合を試行錯誤的に構成することで次第に上部構造が決定していき、内側の集合を構成していくことで下部構造が決定していく(図. 1d参照)。また、モデルから容易に知識を木構造で表現することができる。集合Sを木のノードNSに対応させる。すると、Sの要素は各々NSの子ノードに対応する。このように集合の階層関係は、木の親子関係となり、モデルから自然に木を作ることができる。

(2)モデル表現能力とのトレードオフ

エキスパートシステム構築で常に問題となるのは、知識獲得、つまり専門家のもつ生の知識をシステムで扱えるような形式で取り出す部分である。知識獲得の面から言えば、知識表現モデルは、できるだけ単純であるべきである。その方が、提供者である専門家に分かりやすく、効率的に知識を獲得することができる。逆に対象分野に特徴的な(凝った)推論を実現しようとする知識表現モデルがかなりの表現能力をもたなければならない。その結果専門家にとっては、知識提供時に多くの負担がかかることになる。実際、前述の何ステップかジャンプする推論でさえも、確かに有効であるが、知識を構築していくときに、それを考えるのは不自然というのが専門家の意見である。我々は、このようなことからある程度対象領域の特徴を反映し、かつ知識獲得時に専門家に負担をかけない程度の単純さ(分かりやすさ)をもったモデルの構築を目指した。アプローチとしては、まず考えられる様々な推論を実現するモデルを提案し、専門家により実際のレベルで不必要で複雑と思われる部分を指摘してもらい削るということを繰り返していく方法を採用した。その過程で確信度、評価値(テストにどの

くらいの時間をとるか等)、メタ知識、事象間の原因-結果以外の関係の知識等は削られていった。

3. 3. 知識表現モデルとDBESの機能

DBESが備えた機能で実際のデバッグ作業に特徴的なものを前記のモデルによって説明する。

(1)mark informationとinitial information

推論中に得られた事実や初期情報(initial information)は、モデルではその集合をマークすることに対応する。同じ症状が見い出されたとき、その中の最も内側のマークされたところまでアークが張られる(図2. a参照)。ある初期情報があれば、即座にその事象までジャンプするアークが張られる(図2. b)。

(2)unknown mechanism

モデルではunknown informationはunknownとしてマークされる。フィードバックのとき、推論は、最も内側のマークされた集合まで戻り、そこからまた推論を始める(図3. 参照)。

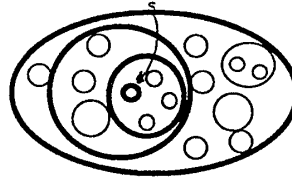


図2a:マークされた集合は推論中に事実として選択されたものである。再びSに到達したとき最も内側のマークされた要素にアークが張られる。

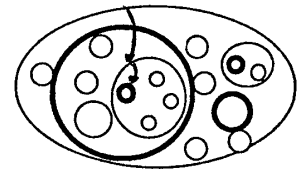


図2b:既知情報はマークされている。アークはこれらの要素を越えて張られる。

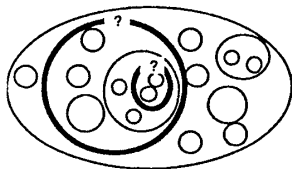


図3a:マークの要素は不確定情報を表す。フィードバックのときは最も内側のマークされたところに戻る。

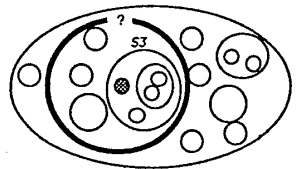


図3b:フィードバック後、事象S3を再分析し原因図が選ばれた。

3. まとめ

交換ソフトのデバッグ作業を反映し、知識獲得においては専門家の知識の整理方法を反映した知識表現モデルを構築した。特に、複雑な推論よりも分かりやすさに重点をおいた。交換ソフトのデバッグのように知識がダイナミックな分野では、専門家自身が知識の保守を容易にできることが必要となる。また、その知識を使ったシステム(ES)は、専門家の自然な振舞いを反映したものでなければならない。今後は、このモデルを共通の言葉として、DBESの機能拡張及び知識エディタの開発を専門家と共同で進めていきたい。

[参考文献]

- [1] Shortliffe, E. J., Computer Based Medical Consultations: MYCIN, Elsevier, NY, 1976.
- [2] Davis, R. and Lenat, D. B., Knowledge-based Systems in Artificial Intelligence, McGraw-Hill, 1982.
- [3] Cunningham, P., Brady, M., Qualitative Reasoning in Electronic Fault Diagnosis, IJCAI'87, 1987.