

要求仕様の理解支援システム

*大西 淳、**山本洋一、*阿草清滋、*大野 豊

*京都大学 工学部、**日立製作所

1. はじめに

要求定義段階において明らかにされるべき性質は何であるべきかについては種々の考え方がある。例えば、エンド・ユーザとのインタフェースすなわち外部から見たシステムの入出力の特性のすべてを明らかにすれば十分であって、その内部構造については明らかにする必要はないという立場もあれば、一方ではエンド・ユーザとのインタフェースのみならずシステムの機能や性能、運用環境、開発費、開発期間なども明らかにすべきであるという立場もある。別の言い方をすれば、前者の立場は要求をエンド・ユーザのビューだけから解析するというものであり、後者はエンド・ユーザ、システム設計者、システム発注者など様々なビューから解析するものである。我々は要求仕様の実現可能性を検証したり、要求仕様からプロトタイプの自動生成を行うことも検討しており、そのためにはエンド・ユーザのビューだけから見た要求仕様では不十分であり後者の立場をとる。

要求仕様は要求分析の結果明らかにされた目標システムとエンド・ユーザとのインタフェースやシステムの機能や性能等を文書化したものである。要求仕様を元にしてシステムの設計は行われるから、要求仕様はシステム開発の依頼者の要求を忠実に反映し、かつ誤り、矛盾、曖昧さを含んではならない。要求仕様の満たすべき特性のひとつすなわちシステム開発依頼者の要求を記述が忠実に反映していることを確認する最も簡便な方法は、依頼者自身に要求仕様を提示し意図通り書かれているか依頼者自身にチェックしてもらう方法である。

ところが、種々のビューから記述された要求仕様をシステム開発の依頼者、例えばエンド・ユーザに見せたとしても、エンド・ユーザは要求仕様を理解できない。エンド・ユーザの興味はシステムとのインタフェースでありシステムの機能構造や開発のために要する費用ではない。さらに形式的な要求記述言語や、システム開発者の専門用語に不慣れであることも要求仕様の理解を妨げる一因となっている。これらの理由から依頼者自身に要求仕様の正しさを確認してもらうことは難しい。依頼者の確認を得ないまま開発を進める場合、製品ができ上がって始めて依頼者の手に渡ることになり、時に依頼者の要求に合わず再開発を行うことになる。

設計者にとっても種々のビューから記述された要求仕様から自分の興味のある部分だけを切り出して読むことは難しい。また要求仕様記述者の知識と設計者の知識との間にギャップがあると要求仕様に書かれた記述者の意図を正確に理解することはできない。

ここでは、人間の理解過程をモデル化し、そのモデルに基づいた要求仕様の説明システムを提供することによって、要求仕様の読み手の知識に応じて要求仕様を正しく理解できる手法について紹介する。この手法を用いることにより依頼者自身によって自分の要求が要求仕様書に正しく反映しているか検証できる。さらには設計者を読み手とした場合、設計者自身もより正しく要求を理解できるため効率良くシステム開発ができる。

2章では、人間のコミュニケーションと理解過程を検討しモデル化する。3章では、検討したモデルに基づいて試作したシステムについて紹介し、システム使用例を示す。

2. コミュニケーションと理解

エンド・ユーザに要求仕様の正しさを確認してもらう過程やシステム設計者に要求仕様を渡す過程は、それぞれ要求記述者とエンド・ユーザもしくはシステム設計者との要求仕様を介したコミュニケーションと考えられる。2.1ではコミュニケーションについてモデル化する。

2.1 コミュニケーション・モデル

コミュニケーションは次のような過程をふんで行われる[1]。

- 1) 知覚作用 (perception)
- 2) 概念化作用 (conceptualization)
- 3) 言語表象との照応作用 (identification with verbal symbols)
- 4) 表出作用 (expression)
- 5) 送信作用 (transmission)
- 6) 視聴覚作用 (audio-visualization)
- 7) 言語表象との照応作用 (decoding)
- 8) 概念化作用
- 9) 了解作用 (comprehension)
- 10) 反応作用 (reaction)

要求分析終了後、要求記述者が要求仕様を書く作業は上の1から4にあたる。要求仕様を読み手が受け取り理解する作業は5から9にあたる。要求仕様をエンド・ユーザが読んでも理解できない理由を次にいくつか示す。

- a. 仕様に使用された語句の意味がわからない。
- b. 仕様に表われるシステムに対する一般的な知識が欠如している。
- c. 自分の興味のある部分とない部分が明確に分離されてなく読みづらい。

理由aはコミュニケーション・モデルの7段階目で、理由b, cは8段階目でコミュニケーションが失敗することにより理解ができないことに対応している。要求の記述者の持つ知識と読み手の知識の間にギャップがあると円滑なコミュニケーションは望めない。2.2節では人間の理解過程をモデル化することによって、ギャップがある場合にどうすれば正しくコミュニケーションできるか検討する。

2.2 理解モデル

システム要求仕様の理解をここでは目的としているため、理解に必要な知識は人間の持つ知識すべてではなくある程度制限される。本節では、知識を一階述語論理を用いて表現した場合の人間の理解過程をモデル化する。

定義1：理解

ある一階述語 L がある人の持っている知識集合から導出可能であるとき、その人は L を理解できる。

定義 2 : 獲得

ある一階述語 L を理解できる場合、もしくは L がある人の持っている知識集合から導出されないが、矛盾を生じない場合、 L は新しい知識として獲得される。

例えば、論理式 A, B, C, D, E に対して $A \leftarrow B, B \leftarrow C$ を知っている時、

$A \leftarrow C$ は理解でき且つ獲得可能である。
 $D \leftarrow E$ は獲得可能である。しかし、
NOT $A \leftarrow C$ は獲得不可能である。

具体例で説明しよう。「どんなプロセスも何らかのデータを受け取る」、「在庫管理というプロセスがある」という知識はそれぞれ

$\text{any } X, \text{some } Y (\text{RECEIVE}(X, Y) \leftarrow \text{PROCESS}(X), \text{DATA}(Y)) . \dots (L1)$
 $\text{PROCESS}(\text{在庫管理}) \leftarrow \dots (L2)$

と表わされる。

これらの知識を持っているならば、「在庫管理は何らかのデータを受け取る」ことは理解でき且つ獲得可能である。なぜならば「在庫管理は何らかのデータを受け取る」という事実は

$\text{some } X (\text{RECEIVE}(\text{在庫管理}, X) \leftarrow \text{DATA}(X)) . \dots (L3)$

と表わされ、この論理式 $L3$ は $L1, L2$ からの論理的帰結であるからである。

「どんなプロセスも何らかのデータを出力する」ことは獲得可能である。「どんなプロセスも何らかのデータを出力する」ことは

$\text{any } X, \text{some } Y (\text{OUTPUT}(X, Y) \leftarrow \text{PROCESS}(X), \text{DATA}(Y)) . \dots (L4)$

と表わされ、 $L4$ は $L1, L2, L3$ のいずれに対しても矛盾しない。我々の定義では矛盾しない (not inconsistent) 論理式は獲得可能であるとする。

しかし、「在庫管理はデータを受け取らない」ことは獲得不可能である。「在庫管理はデータを受け取らない。」ことは

$\text{any } X (\text{NOT RECEIVE}(\text{在庫管理}, X) \leftarrow \text{DATA}(X)) . \dots (L5)$

となり、 $L1, L2$ と矛盾する。我々は矛盾する (inconsistent) 論理式は理解も獲得も不可能と定義する。

$L1, L2$ を知識として持つ人に「在庫管理は何らかのデータを出力する」ことを理解させるには $L3$ を獲得させればよい。このようにある事実をある人に理解させるためには、その事実を論理的帰結とするような論理式のうちその人が持たないものを獲得させることによって理解させる事ができる。2. 3 節では要求仕様の理解に必要な知識を明らかにする。

2. 3 要求仕様の理解

要求定義過程モデルを図1に示す。要求記述者は様々なシステムに対する概念「対象システムに関する知識」と目標システムに依存しない概念「システムに関する一般的な知識」を持つものとする。。エンド・ユーザの発言を記述者は解析し正しい発言をエンド・ユーザの要求として得る。「対象システムに関する知識」とエンド・ユーザの要求とから目標システム要求プロトタイプを選択し、「システムに関する一般的な知識」と「要求仕様記述言語に関する知識」を用いて要求仕様を作成する。ここでシステムに関する一般的な知識は「プロセスは何らかのデータを受け取り何らかのデータを出力する」と言った知識を意味する。また対象システムに関する知識は、例えばいま対象システムを在庫管理システムとした時「在庫管理システムは在庫検索と在庫更新機能が必要である」と言った対象システムに固有な知識が相当する。これらの知識は一階述語で十分記述可能であり、計算機中に格納する。これらの知識を持たない人に対して、計算機から適切な知識を獲得させることによって要求仕様を理解させる。言語に関する知識は2. 1節で示したコミュニケーション・モデルの第7段階目の言語表象との照応の際に必要となる。日本語を知っている人なら、少なくとも日本語で書かれた要求仕様の字面は追えるが、日本語を知らない人には日本語要求仕様は皆目理解できない。ここでは要求仕様記述言語として日本語を採用することにより日本人ユーザは持っている知識とみなし、計算機に持たせることはしない。

従って我々の理解モデルに基づくと要求仕様を理解するために必要な知識として

- 1) システムに関する一般的な知識
- 2) 対象システムに関する知識
- 3) 読み手の持つ知識

があれば十分である。

本研究では読み手としてエンド・ユーザもしくは設計者を想定しているが、以下のように読み手の持つ知識を定める。

- ・ エンド・ユーザはここでは要求の発言者でもあると仮定しており、彼の持っている知識は要求仕様の記述者に彼自身が発言したすべての事柄と限定する。
- ・ 設計者の知識はシステムに関する一般的な知識のすべてと定め、対象システムに固有な知識は欠如しているものとする。

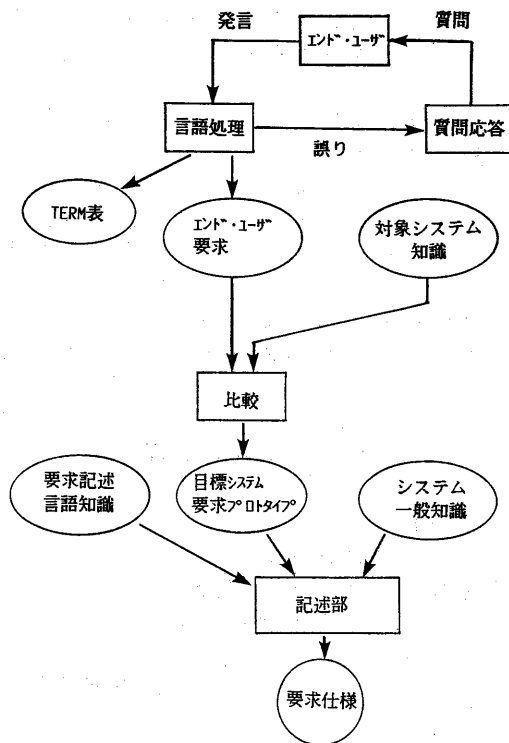


図1. 要求定義過程モデル

さらにエンド・ユーザの興味はシステムと外部とのインタフェースのみとしてあらかじめ要求仕様から外部とのインタフェース部分の仕様を切り出しておく。一方設計者の興味は機能構造, 入出力インタフェース, データ構造など種々考えられる。

我々は設計者に切り出しのためのビュー記述言語を提供しビューを設計者自身に記述してもらうことによって設計者の望むままの切り出しを可能にする。

以上の検討に基づいた理解支援概要を図2に示す。理解支援システムのユーザは要求仕様中で判らない部分を質問する。質問された要求仕様に対応する一階述語を定理証明系で処理することによって、証明に使われた知識が明らかになる。ユーザの持つ知識をあらかじめマーク付けしておくことにより証明に使われた知識のうちユーザの持たない知識が判明する。最後にユーザが持たない知識を獲得させることによって要求仕様をユーザに理解させることができる。

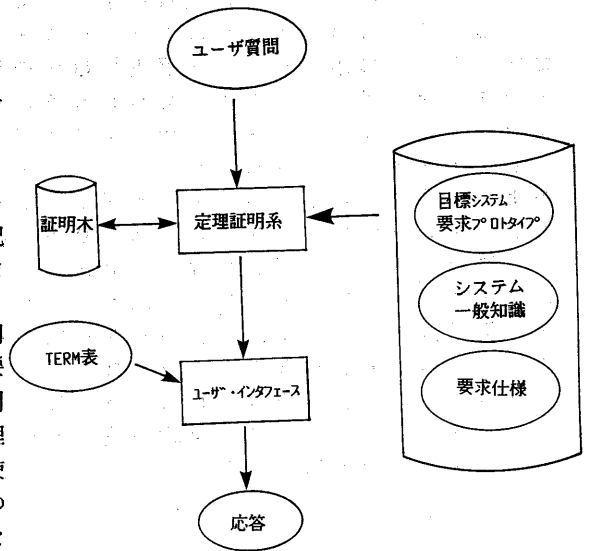


図2. 要求仕様理解支援

3. 要求仕様理解支援システム SURE [2]

3. 1 システム構成

SURE (Supporting system for Understanding REquirements specification) は図3に示した構成をとる。変換モジュールはエンド・ユーザが発言した部分にマーク付けされた要求仕様を一階述語に変換する。SUREの核はREX (Requirements EXplainer) と呼ばれる推論プログラムである。REXは図中の4種類の知識を取り込み、SUREユーザからの質問に答えることによって要求仕様の理解支援を行う。REXからの応答はユーザ・インタフェース部の知能端末上で漢字に変換される。REXは日立のM240-H上でProlog/KRにより作成された。変換モジュールはCにより実現されている。ユーザ・インタフェース部は日本電気のPC-9801F上でCにより実現されている。

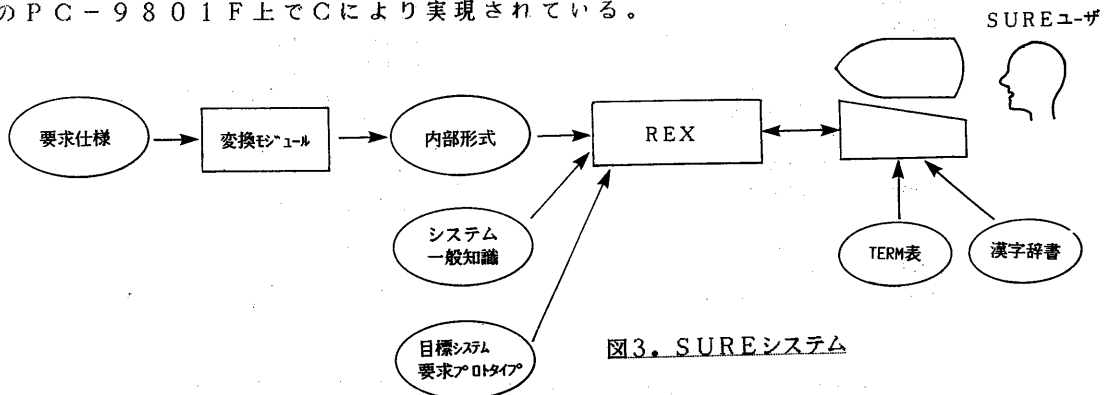


図3. SUREシステム

SUREは我々の理解の定義に基づいてユーザからの質問に対して、そのゴールをユーザの知識と定めて推論を行う。理解支援はその推論のパスを順次示すことによって行われる。例えばプロセスAがエンド・ユーザの発言に表われ、要求記述者はAをさらにB, C, . . . , Xのように分割したとする(図4参照)。ここでエンド・ユーザであるSUREユーザがXについて質問すると、SUREはプロセスAから始まりXに達するまでのパスを提示する。またパスの提示途中でその推論が行われた理由を尋ねる事も可能である。例えばプロセスAがBとCに分割された理由も尋ねられる。さらにエンド・ユーザの用語と記述者の用語が異なる場合、エンド・ユーザの用語に直して提示される。

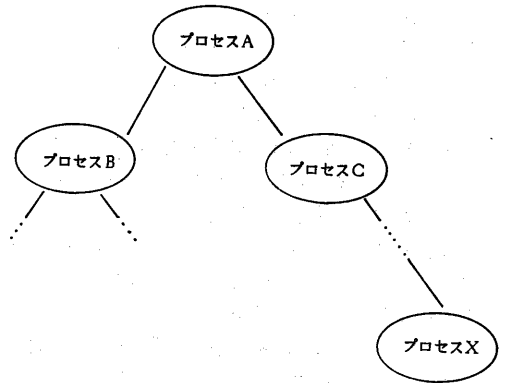


図4. プロセス構造例

3. 2 例

図5に示したエンド・ユーザの要求に対する要求仕様の一部を図6に示す。図7にエンド・ユーザ向きに切り出した要求仕様の一部を示す。この切り出された要求仕様に対する質問の応答例を図8に示す。質問以外にユーザのコマンドとして、肯定を表わすY(es),理由を尋ねるW(hy),他の説明を求めるA(nother),終了を表わすQ(uit)がある。図9には設計者によるSURE応答例を示した。

商品の在庫管理をしたい。商品の在庫数を知ることができ、かつ入庫出庫の際在庫数の変更が出来なければならない。また商品の単価も知りたい。どの商品かを見分けるには、商品名とその商品の色、太さ、入っている線の数、材質を使う。

図5. エンド・ユーザからの要求

| | |
|-------|-----------|
| 入力 | 要求は |
| 問いただし | と |
| 更新要求 | を部分として持つ。 |
| 要求 | が入ると |
| 応答 | が返される。 |
| 問いただし | が入ると |
| 在庫数 | が返される。 |
| 出力 | 応答は |
| 在庫数 | と |
| 更新応答 | を部分として持つ。 |

図7. エンド・ユーザ向きに切り出された要求仕様の一部

| | |
|----------|-----------|
| プロセス | 在庫管理は |
| 要求 | を受け、 |
| 応答 | を返す。 |
| 在庫調査 | と |
| 在庫更新 | を部分として持つ。 |
| 入力 | 要求は |
| 検索要求 | と |
| 更新要求 | を部分として持つ。 |
| プロセス | 在庫調査は |
| 検索要求 | を受け、 |
| 在庫数 | を返す。 |
| 入力解析 | と |
| 商品ファイル検索 | を部分として持つ。 |
| 出力 | 応答は |
| 在庫数 | と |
| 更新応答 | を部分として持つ。 |
| プロセス | 在庫ファイル検索は |
| 検索条件 | を使い、 |
| 在庫数 | を返す。 |

図6. 要求仕様の一部

?Q 要求という入力は何か

要求は問いただしと更新要求を含んでいます。

?Q WHY

ユーザは要求において二つの行動が可能です。
一つは在庫の検索で問いただしにより行います。
もう一つは在庫の更新で更新要求により行います。

要求は問いただしと更新要求を含んでいます。

?Q YES

?Q 問いただしという入力は何か

問いただしはコマンドとパラメタ色、太さ、
入っている線の数、材質からなります。

?Q WHY

問いただしはコマンドとパラメタ色、太さ、
入っている線の数、材質からなります。
これはあなたの要求の一つです。

?Q ANOTHER

問いただしはコマンドとパラメタ色、太さ、
入っている線の数、材質からなります。
これ以上の説明はありません。

?Q QUIT

図8. SUREのエンド・ユーザによる応答例

?Q 商品ファイル検索というプロセスは何か

在庫管理をおこなうプロセスがある。
これは、エンド・ユーザの要求の1つです。

?Q YES

在庫管理は在庫調査と在庫更新という部分に分かれます。

?Q WHY

在庫管理は現在の在庫を調べる部分—在庫調査と
入荷または出荷で在庫に変化があったとき
在庫を変更する部分—在庫更新からできています。

在庫管理は在庫調査と在庫更新という部分に分かれます。

?Q YES

在庫調査をおこなうプロセスがある。

?Q WHY

在庫はプロセスであり
それが分かれた在庫調査もまたプロセスでなければならない。

在庫調査をおこなうプロセスがある。

?Q ANOTHER

在庫調査は入力解析と商品ファイル検索という部分に分かれます。

?Q WHY

在庫調査のためには何について調べるのか判断して—入力解析し
そのあと在庫を調べます—商品ファイル検索。

在庫調査は入力解析と商品ファイル検索という部分に分かれます。

?Q YES

商品ファイル検索をおこなうプロセスがある。

?Q QUIT

図9. SUREの設計者による応答例

4. 結び

本稿では人間の理解過程をモデル化し、システム開発者に要求仕様を理解させる手法を示した。エンド・ユーザが要求仕様を理解することによって、外部とのインタフェースが正しく定義されているかエンド・ユーザ自身にチェックしてもらうことが可能になる。プロトタイピングも外部とのインタフェースの確認に有効であるが、エンド・ユーザ自身は意識しなかったコマンドやモードがなぜ必要になるかを理解することはできない。我々の手法では要求記述者がなぜコマンドを用意したりモードを設けたかをシステム一般の知識として用意することによりエンド・ユーザからの質問に答えることができる。

また設計者が自分の興味のある部分を切り出して理解することにより正しいソフトウェア開発が可能となる。設計者にとってもシステム一般の知識は持っていてもシステム固有の知識を持っているとは限らないので、なぜ要求記述者があるプロセスを定義したのかわからないかもしれない。我々の手法ではシステム固有の知識を計算機に持たせることによって設計者からの疑問に答えることができる。

提案した手法に基づく試作システムの効果を評価中である。

[参考文献]

1. 平井昌夫「ことばの百科事典」 三省堂
2. Y.Yamamoto: "A Supporting System for Understanding Requirements Description.", Master thesis, Kyoto University, Feb., 1984
3. 大西 淳他「要求仕様記述の理解支援」
情報処理学会第28回(昭和59年前期)全国大会予稿集2K-3
4. 山本洋一他「要求仕様記述の視認性向上」
情報処理学会第28回(昭和59年前期)全国大会予稿集2K-2
5. A.Ohnishi et al.: "Improvement of Communicability of Requirements Description", Proc. of International Symposium on Current Issues of Requirements Engineering Environments, Sept., 1982
6. 後藤滋樹「推論と問題解決」
電子通信学会誌 Vol.65 No.4, 1982年4月
7. 斎藤正男他「知的情報処理の設計」
コロナ社 1982年11月
8. 数理科学 特集「知識構造」
サイエンス社 1983年6月