

4F-2

コンテキストを導入した要求定義の知的支援

荒木 誠 大西 淳 阿草 清滋 大野 豊

(京都大学 工学部)

1. はじめに

要求仕様書は発注者の要求を正確にシステム設計者に伝達する必要があるが、不相当であったり、実現不可能であってはならない。従来の形式的な要求言語は、文法に厳しい制約が課されることが多く、記述量が膨大になったり、また簡明さに欠けることもあり、ユーザーフレンドリなインターフェースを必ずしも提供してはいない。

本稿では、要求記述におけるコンテキストを抽出し、また情報システムに関する一般的性質を知識として用いることで、人間にとってより自然な形で要求の記述が可能とするような支援手法について述べる。

2. 要求定義の知的支援手法

2.1 概要

本手法による要求定義の支援は、2つのフェーズから成る。まず第一のフェーズでは、入力される要求記述のコンテキストを抽出し、これを利用して要求記述の解釈を行う。すなわち、本手法で採用した格文法に基づく要求言語では、本来一文で意味的に独立したものであり、任意の順序で要求文を記述することができるが、要求は通常意味的にまとまった順で捉える方が書き易い。反面、指示代名詞や省略用法などを用いた記述ではあいまいさの生じる恐れがある。そこで、要求記述者の現在記述している主題、あるいは脈絡といった動的に切り替わる情報をコンテキストとして抽出し、要求記述中のあいまいな部分の解釈に利用する。但し、要求分析が不十分なために生ずる要求記述のあいまいさは計算機によって解釈することはできない。そこで、本手法では支援すべきユーザとしての要求定義者について次の仮定を設ける。

「記述すべき要求はあいまいなく完全に理解し、また要求記述言語についても理解しているが、冗長な要求記述を書く手間を省きたい。」

つまり、記述者本人にとってはあいまいではないが、別の人が一部だけを見るとあいまいであるような部分について解釈を行う。

第二のフェーズでは、全ての要求記述が入力され、

第一のフェーズで単文化された後、情報システムに関する基本的性質を知識として用い、要求文単位での矛盾を検出し、抜けを推論によって補う支援を行う。以下、第一のフェーズを2. 2で、第二のフェーズを2. 3でそれぞれ説明する。

2.2 コンテキストを利用した支援

要求定義段階でのコンテキストがいかなるものであるべきかを考察するために、まず要求定義を定義者と計算機の対話として捉え、コンテキストを必要としないコミュニケーションの条件を考察する。始めに、次の2つの集合を定義する。

C : 記号表現の集合

M : 記号内容の集合

伝達しようとする概念は、記号内容によって構成される。次に、概念が誤りなく伝わるためには、CからMへの写像を行う関数fについて考慮する必要がある。

$$\forall c_i \in C, \exists m_i \in M$$

such that $f : c_i \rightarrow m_i$

ここでCからMへの関数は多対一の対応で定義され、これは同義性として解釈することができる。この対応は、一対多(多義性と解釈される)や多対多であってはならない。なぜなら、単一の c_i に対して複数の解釈が与えられることになってしまい、この場合は誤りなく概念を伝達することができない。最後に、 c_i の許容される結合としての、構文規則に関する規定が必要である。

つまり、コンテキストを必要としないコミュニケーションの条件は次の4点から成っている。

- 1) Cに関する規定
- 2) Mに関する規定
- 3) f (CとMの対応)に関する規定
- 4) c_i の結合に関する規定

これらの条件を、要求定義において誤りなく概念が伝達されるための条件として解釈すると、次のようになる。

- 1) 要求文作成の場合に要求定義者が用いることのできる要求文構成要素Cの表現が明確に規定されて

いること。

- 2) それぞれの要求文構成要素に担わせうる情報がセマンティックスMとして明確に規定されていること。ここで、動詞のセマンティックスを格フレーム [1]、名詞のセマンティックスを動詞の格フレームから導かれるタイプと定義する。
- 3) 規定されている要求文構成要素の表現CとセマンティックスMの対応fは、常に一対一、あるいは多対一であること。同義性は許されるが、多義性によるあいまいさが生じる可能性があってはならない。
- 4) 要求文の構成要素である動詞や名詞の結合に関しては、許容される結合が全て規定されていること（そこから逸脱した結合で用いられることはない）
これは、C、Mに制限を持ち、構文に制約の課せられる要求定義環境として捉えることができる。すなわち、要求記述に現れる記号と計算機内部で概念化した記述との関係は多対一であり、対応をとって考えることができる。

これらの条件から逸脱した際にコミュニケーションを取り持つのがコンテキストである。前記条件からの逸脱とは、具体的には次に示すものである。

- 1) 対応するmが存在しないcが現れた
- 2) 対応するcの存在しないmを表現したい
- 3) 使われたcが複数のmに対応する
- 4) cの並び方に関する制限からの逸脱

これらを要求定義段階での条件からの逸脱として捉えると、まず1)については解釈が本質的に不可能であり、また2)についても、2. 1で示した要求定義者の仮定から、考慮しなくて良いと言える。結局、3)と4)の逸脱が考えられる。3)については、代名詞や「その他の」等の単語を新たにCに加えることにより、対応するMが複数の候補を持つような場合であり、4)は記述に省略部分があるような場合である。

これらを解釈するには、例えば代名詞ならば以前の記述における同じタイプの名詞から、また集合演算の必要な記述についても以前記述に現れたの各タイプの名詞の履歴をとっておけば解決できる。つまり、前述の条件から逸脱したために起こるあいまいさを解消するために必要な動的に変化するデータの集合を「要求定義におけるコンテキスト」と呼ぶ。このように動的に移り変わるコンテキストを抽出し、解釈に利用することで、従来単文単位でしか扱うことのできなかった要求記述を複文や重文、あるいは省略用法や指示代名詞を用いることができるようになり、要求定義者にとってより自然な形での記述が可能になる。

2. 3 暗黙推論による支援

コンテキストを利用する記述支援が動的な知識を利用するのに対し、本フェーズでは情報システムが持つ基本的性質という静的な知識（ルール）を用いて、要求記述の矛盾を検出し、また暗黙で決定し得る部分については推論によって自動的に記述を補おうとするものである。以下に一例を示す。

要求文：

- 1) プロセスPは子プロセス P1, P2 より成る
- 2) データAは A1, A2 より成る
- 3) データBは B1, B2 より成る
- 4) プロセス P1 は A1 を入力する
- 5) プロセス P2 は A2 を入力する
- 6) プロセス P1 は B1 を出力する
- 7) プロセス P2 は B2 を出力する

知識（ルール）：

子プロセスの入力データの集合は親プロセスの入力データを分割したデータ集合に一致する

子プロセスの出力データの集合は親プロセスの出力データを分割したデータ集合に一致する

入力データの階層とこれらを入力するプロセスの階層は一致する

出力データの階層とこれらを実行するプロセスの階層は一致する

導出される要求文：

- 8) プロセスPはデータAを入力する
- 9) プロセスPはデータBを出力する

このようなルールはデータフロー、プロセスやデータの構造、制御の受渡しなどについて約20種類用意されており、要求定義者の労力を低減させるのに有効である。

3. おわりに

要求定義におけるコンテキストを抽出し、利用することで、また情報システムに関する基本的性質をルールとして用いることで、人間にとってより柔軟な記述が可能である要求定義環境を作り出すことができる。現在、要求言語としてJRDL [1]を採用し、本手法によるプロトタイプを作成してその有効性を評価中である。

参考文献

- [1] 大西 淳, 阿草清滋, 大野 豊
要求モデルに基づいた要求定義支援手法
情報処理学会「プロトタイプと要求定義」
シンポジウム, 昭和61年4月