



**《SLPの基本的な考え方》**

ソフトウェア要求仕様書はプログラムが実現すべき内容からなる文の集まりです。文は内容の説明文も含まれますが、最終的には実現すべきことが「～ならば、～せよ、そうでないならば、～せよ」という条件付き命令文と、「～せよ」だけの無条件命令文からなります。「～」は文で、主語や目的語（対象物）と述語（対象物の状態）からなります（単位文）。「ならば」と「そうでないならば」を条件接続詞と呼ぶこととします。すると、要求仕様書は単位文と条件接続詞で記述される、といえます。

**《SLP構文》**

- 2つの基本構文と3つの派生構文と、2つの参考文からなります。これらをSLP構文と呼びます。基本構文は、例示のDo文とif文です。派生構文は基本構文を基に、単位機能を表すFn文、if文の派生としてのswitch文、繰り返しのloop文です。参考文はコメント文と改行文です。
- 単位文はSLPを記述するための基本単位となります。単位文は「<対象物>は{ある状態を持つ}」という陳述形式や、「<対象物>を{ある状態にせよ}」という命令形式で表現されます。対象物の名前を「メンバー名」と呼ぶこととします。このメンバー名は単位文の主語や目的語に相当します。対象の状態を「状態名」で表します。状態名は陳述述語や命令述語に相当します。たとえば「実行ボタンがONであるならば」は「if <実行ボタン>が{ON}である」と記述します。「if」に対しては必ず“else”が付帯され、if-else-endifの構造を持ちます。入れ子の構造です。メンバー名の状態を設定（命令）する時には、「Do<実行ボタン>を{OFF}せよ」と記述します。
- 要求仕様内容は階層構造的に表現できます。大きな文書も内容の親子・兄弟関係が明確にされます。階層構造化される単位を単位機能と呼び、SLP構文で論理的に記述されます。これにより文が矛盾無く記述されているか、また用語が文書全体の中で統一されているかを検査できます。このことは、製品企画、要求分析者、開発者、テスト者等の思い違いを激減させます。納期や開発予算を最適化します。

**仕様書例** (“SCADE”のテキストより)

【クルーズ制御仕様】

- クルーズオン制御
- クルーズ制御状態がオフで、ONボタン又はResumeボタンが押され、車速が制御可能範囲内で、アクセルペダルとブレーキペダルが踏まれていない時
- クルーズ制御状態がスタンバイで、車速が制御可能範囲内で、アクセルペダルが踏まれていない時
- クルーズオフ制御
- 初期状態は必ずオフである
- ブレーキペダルが踏まれるか、OFFボタンが押された時
- クルーズスタンバイ制御
- クルーズ制御状態がオフで、ONボタン又はResumeボタンが押され、車速が制御可能範囲内でない又は、アクセルペダルが踏まれて、ブレーキペダルが踏まれていない時
- クルーズ制御状態がオンで、車速が制御可能範囲内でない又は、アクセルペダルが踏まれている時

実際にお試しください。ダウンロード→<http://www.jfp.co.jp/slp/>

例はよく書かれた仕様です。しかしSLPで要求分析をしてみると、仕様には書かれていないことも知らないと記述できない所があることがわかります。このことは、書き手は仕様の意味をすべて知らなければならないことを物語っています。しかし、すべてを知らないからといって臆することはありません。SLPは「形式」と「意味」の違いを明確にした上で、「意味」にアプローチする方法を提供しています。このことは仕様の受け手は「形式」さえ確かであれば、少なくとも最初のうちだけ新たな領域の知識を持たなくても許されるという根拠を示しているように思われます。

```

if <クルーズ制御状態>が[オフ]
  if <ONボタン又はResumeボタン>が[押された]
    if <車速>が[制御可能範囲内]
      if <アクセルペダルとブレーキペダル>が[踏まれていない]
        Do<クルーズ制御>を[オン]せよ
        (:上のifの4つの重なりは、例題1の項番5)
      else
        if <アクセルペダル>が[踏まれている]
          if <ブレーキペダル>が[踏まれている]
            Do<クルーズ制御>を[スタンバイ]とせよ
            (:このケースは、例題1の項番5の後)
          else
            Do nothing
          endif
        else
          Do nothing
          (:ここでは、上の場合分けからして、<アクセルペダル>が[踏まれる]かつ)
        endif
      endif
    else
      Do nothing
      (:ここでは、上の場合分けからして、<アクセルペダル>が[踏まれる]かつ)
    endif
  endif
else
  (:<車速>が[制御可能範囲内]のケースである)
  Do<クルーズ制御>を[スタンバイ]とせよ
  (:このケースは、例題1の項番5の前半「、)
endif
else
  (:Pending)
endif
else
  if <クルーズ制御状態>が[スタンバイ]
    if <車速>が[制御可能範囲内]
      Do<クルーズ制御>を[オン]せよ
      (:上のelseとifの3つの重なりは、例題1の)
    else
      Do<クルーズ制御>を[オフ]とせよ
      (:このケースは、例題1の項番4の前半「ブ)
      (:Pending)
    endif
  else
    Do nothing
    (:Pending)
  endif
else
  if <クルーズ制御状態>が[オン]
    if <ブレーキペダル>が[踏まれる]
      Do<クルーズ制御>を[オフ]とせよ
      (:このケースは、例題1の項番4の前半「ブ)
    else
      if <OFFボタン>が[押された]

```

**自然言語**で書かれた要求仕様書をSLPで記述することによって、要求仕様書は論理整合性（論理無矛盾、条件網羅性）と一意性（用語統一、意味明確）を獲得することができます。

記述はSLP構文で単位文を書くことで行います。また複数の単位文をまとめ、単位機能として階層構造化できます。

記述は構文のメニュー選択で行います。次行に記述可能な構文をガイドします。さらに以前に書いたメンバー名や状態名を選択するなど、ユーザビリティを高めています。

SLP構文はif-else-endif構文により条件網羅性と形式無矛盾を保証します。用語の登録により、意味矛盾も検査されます。これらにより文書の論理無矛盾（形式無矛盾、意味無矛盾）を保証します。

SLPでは単位文の指す意味を単位意味と呼びます。レビューも単位文ごとにできます。その場合、SLPでのレビューは単位意味ごとに仕様を確認することになりますので、レビューにありがちな細かな確認ミスを防止できます。

**仕様変更**は開発にはつきものです。仕様変更とは、SLP的にいえば「単位文の用語を変えること、あるいは単位文同士の接続関係を変えること」といえます。SLPでは各々の単位文は仕様書全体に渡って関連づけられていますから、用語や接続関係を変更することによる影響範囲も分かります。変更を確実に効率的に行うことができます。

仕様変更の差管理も行うことができます。if-else-endif構造を決定表に変換し、条件の一覧性を提供します。またテストの仕様原案を提供します。

SLP構文で記述した文書を通常文に逆変換し、条件の網羅性や用語の統一を自然言語に近い表現で行うことができます。要求の識別子を指定することで、要求のトレーサビリティを実現できます。

右は左のように階層化をせず、全て単位文で記述したものです。普通は左のように書き、要求内容をいくつかつかにまとめて分類整理します。

文はメンバー名<、状態名{}から構成されています

メンバー名や状態名は記述の一覧から選択可能です

助詞、助動詞を<>や{}の後ろに記述できます