JFP
metaphor Lab.

SLP 記述ガイドライン

SLP_NLX 版 2.3.3_R

1 はじめに

この文書の目的は、ツール SLP（SLP_NLX 版）を使用するための、記述ガイドです。記述の対象は要求や要求仕様です。ここで、要求仕様とは要求を実現するための詳しい記述内容という程度の意味です。SLP では、要求と要求仕様の区分にこだわらず、両者を広く「要件」と呼んでいます。よって、「要求項目」、「要求事項」なども「要件」と呼んでいます。

想定される読者は、要件を記述する人、また管理する人です。

SLP は要件を記述するために固有の構文を用いています。SLP 構文と呼びます。SLP 構文は論理的観点から記述を促す構文で、" Syntax from Logical Point of view"の略です。
" NLX"は" Natural Language eXtension"の略で、自然言語で書かれた文（「自然文」：natural sentences）を SLP 構文に変換する機能を意味します。ただし、対象は日本語です。

SLP の目的は要件を正確に記述し、読み手に誤解なく、要件の意味を伝えることにあります。そのために、まずは構文をそろえて、構文により論理的な検査ができるようにしています。ここで論理的とは、文の論理矛盾や語句の検査ができることを指します。

SLP は使いやすさを考慮し、自然文で記述することとしています。しかし、自然文は曖昧さを誘発するとされ、使いやすさの一方で警戒もされます。そこで、SLP では自然文をあらためて SLP 構文を記述することを行う必要があります。検査はこの SLP 構文に対して行われるからです。

SLP 構文の記述は、つい曖昧になりがちな自然文を校正する機会とみることもできます。ただ同時に自然文の記述の検査も含め、自然文の SLP 構文への自動変換機能を実現しています。先の" NLX"です。「自動構文化」と呼んでいます。ただし、どんな記述も自動構文化できるわけではありません。自然文も通常の国語文法が乱れていれば、「自動構文化」機能もお手上げです。このような場合には、書き手に記述の修正を求めます。結果的には書き手の自然文も改善されます。

また、要件は多くの章や節、項からなります。これにより、要件全体の「概念（考え）」を分類し、上下に配置し、概念を統合的に、また整合的に表現することが求められます。いわゆる" Integrity"です。

SLP は分類と上下の配置を「階層化」と捉え、SLP の中で階層化が表現できるようにしています。この階層化も一つの構文とし、先の SLP 構文の中の 1 つと扱っています。よって、SLP 構文は、要求仕様文書の自然文の曖昧さを除くだけではなく、概念の統合・整合性、つまり" Conceptual Integrity"を高めることを促します。

1.1 この文書の構成

このガイドラインでは第 2 章で SLP の使い方を説明します。
2.1 節では SLP 文書を作成する方法を説明します。要求を構文規則に従って記述するアドバイスもします。
2.2 節では既存の SLP 文書を編集する方法を説明します。
2.3 節では、SLP 文書を分析することで、要求仕様書の品質向上に役立てるためのヒントを示します。
第 3 章では SLP をソフトウェア開発にどのように応用していくかを説明します。
3.1 節では、要求仕様書を最初から SLP で作成する方法を説明します。
3.2 節では、Microsoft Word や同 Excel などで書かれた要求仕様書と、SLP との併用法を説明します。
3.3 節では、既存の要求仕様書を修正し SLP を用いて新しい要求仕様書を作成する方法を説明します。
3.4 節では、SLP 文書と他の文書との対応関係をトレースする方法（トレーサビリティ法）を説明します。

3.5 節では、主要な形式手法 VDM（Vienna Development Method）と SLP との関係を簡単に触れます。
3.6 節では、SLP を用いてテスト計画を作成する方法を説明します。

1.2 アドバイスの略語

この文章には、通常の説明文に加えて、その簡潔な要約である「略語」が含まれています。アドバイスには、利便性のため、連番の略語を作成しました。略語はグループごとに接頭語を付けて区別しています。各グループの接頭語を以下に示します。

接頭語	意味
NM	名前の付け方
LD	論理記述欄を記述する方法
FN	SLP の機能の紹介
GN	全般的な心構え
UI	操作方法
AS	自動構文化

また、巻末にはアドバイスの一覧を収録しています（第 4 章）。

1.3 SLP とは

SLP は、要求仕様書の用語の統一と、文の論理整合性をはかることによって、要求仕様に関する多くの問題の解決を目指します。
ここで用語の統一とは、要求文を構成する語句を整理し分類することを指します。ここで語句とは、物事を指示する対象と対象に関する属性を指します。SLP では、対象と属性からなる文を単位文と呼びます。対象と属性を、それぞれをメンバー名と状態名と呼びます。メンバー名と状態名が整理され分類されたとき、これらの語句は統一的なものとなります。

簡単な図示：単位文＝｛対象，属性｝＝｛メンバー名，状態名｝

また、論理整合性とは、単位文の間に矛盾がないこと、また条件に抜けがないことを指します。
要求仕様に対するこのような考えは、要求仕様を文の集合としてとらえるところにあります。そして、文の集合は文同士が含意や連言などの論理関係子で結ばれ、個々の文は単位文から成ります。さらに、単位文と論理関係子から構成される文を論理関係文と呼ぶこととします。

要求仕様文書＝｛文の集合｝＝｛文，｛含意，連言，，｝｝＝｛文，論理関係子｝＝｛単位文，論理関係子｝＝｛論理関係文｝

このように単純化することにより、文や語句は比較の対象となります。文や語句は比較され、意味が精義化（refine）されて行きます。
最終的に無矛盾で語句が一意である要求仕様を手にすることができます。

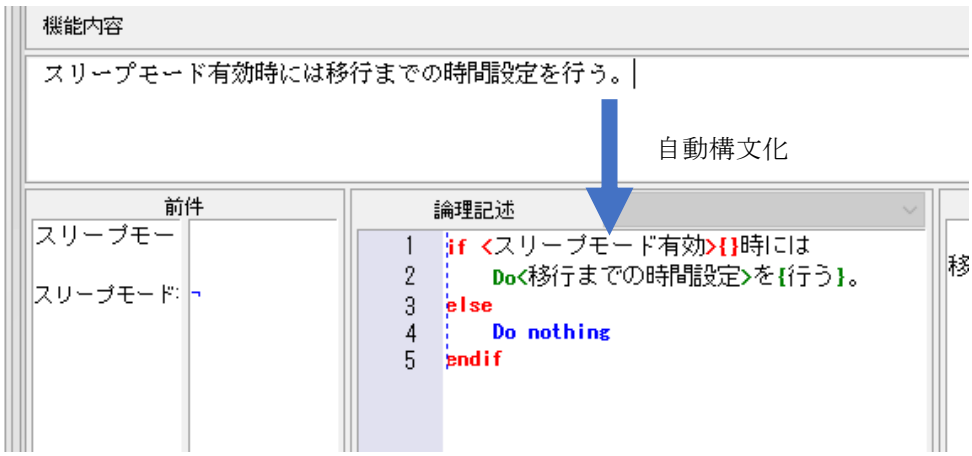
1.3.1 要求の構文化手法

要求開発には下のように４つの工程があります。開発プロセスは前後、またそれ以前の工程と行き来しながら進みます。「・」で示したアクティビティは各工程に「主に」分類されもので、他の工程にもまたがって存在します。

１．要求獲得：	・要求を様々挙げる。 ・図や表なども用いる。 ・要求項目別に分類整理する（章立てする）。
２．要求分析：	・関連する要求が他に無いか明確にする。 ・要求の意図、理由を明確にする。 ・非機能要件を明確にする。
３．要求仕様化	・要求や機能の意味を考え、意味同士を関係づける。 ・意味を最小単位の文（単位文）にする。 ・文を条件に応じた論理関係（含意、連言…）で結ぶ。 ・文の用語を統一する。
４．要求妥当性確認：	・すべての要求の論理関係（条件）を確認し、要求の意味が実現性や採算性等からも適切かを確認する。

1.3.2 構文化手法の自動化

手動で行う構文化に自動化を導入しました。先述の「NLX」を指します。
これは機能内容欄に書いた自然文による要件を当ツールの構文に自動変換するというものです。簡便な記述規則に従うだけで高い変換効率が得られます。またこの記述規則はチームの要件の書き方規則を最後まで維持する効果を持ちます。



詳しい「自動構文化」の説明は、別途『操作説明書』を参照してください。

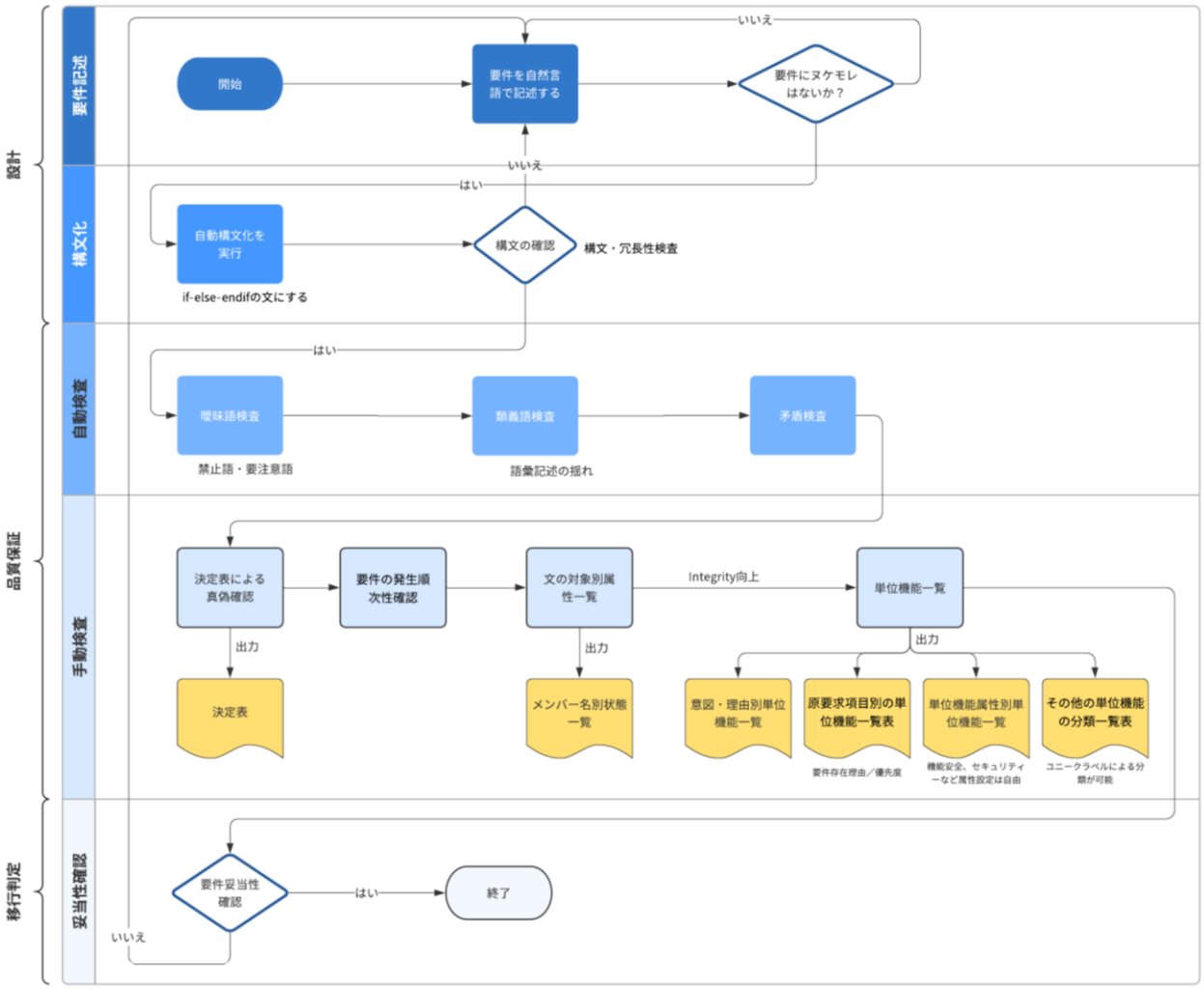
なお、自動であれ、手動であれ、作成すべき構文は、後述の通りです。
自動で作成したものは、確認が必要です。
手直しもあるかと思いますが、手直しは以下の構文の説明に従って行ってください。

2 SLP を使う

SLP の使い方について説明します。操作や各ダイアログの説明については「操作説明書」を参照ください。

2.1 SLP_NLX ワークフロー

SLP_NLX を使ったワークフロー図を以下に示します。



2.2 SLP 文書を作成する

SLP で文書を作成します。

2.2.1 SLP 構文で要求を記述する

要求は先に示したように論理関係文の集合です。この論理関係文の任意の集まりは、諸要求の集まりです。
これを SLP では、要求とか要求事項、また要求項目とは呼ばず、機能とみなします。また、ひとかたまりという意味で「単位」という接頭語を付けることとします。つまり、「単位機能」です。
また「機能」と呼ぶ他の理由は、いわゆる「非機能要件」も含むからです。
以上より、SLP 文書は複数の「単位機能」の集まりです。
ツールでは、この単位機能を書く欄が２つあります。
１つは機能内容欄です。ここでは普段われわれが用いる通常の文で要求の記述をします。通常の文は自然文（natural sentences）とも呼ばれます。
もう一つの欄は、論理記述欄です。機能内容欄に書いたものと同じ内容のものを、論理的に記述します。先に述べた論理関係文です。
この論理関係文の記述のためには規則があります。構文規則と呼びます。ただし、ツールではこの規則をガイドしますので、ユーザーは語句（メンバー名、状態名）を記入するだけで構文を作成でます。もっともどの構文を選ぶかはユーザーが決めないといけません。

2.2.1.1 論理記述欄の記述方法

論理記述欄に論理関係文を書きます。
構文の選択は右クリックメニューで行います。構文規則はツールがすべて提供します。
語句の入力は毎回入力するのではなく、すでに書いたものは、選択可能になっています。語句の揺れを防ぎます。
ツールが提供する構文規則や語句の揺れ防止などの支援機能も含めた機能の全体を「SLP 文法」と呼ぶこととします。

FN-01 SLP 文法で要求を記述するには論理記述欄を用いる。

UI-01 論理記述欄に構文を追加するには右クリックメニューを用いる。

2.2.1.2 構文の意味と使用法

SLP 文法の構文のうち、要求を記述するために不可欠なものは「if」「Do」「Fn」の 3 種類です。それ以外に、用途によっては便利な構文が、「switch」「Do nothing」「コメント」「改行」「loop」「exitlp」「for」「while」の 8 種類です。
なお、これらの構文は、他の記号に変えることができます（「環境設定」参照）。

LD-01 「if」「Do」「Fn」の 3 種類の構文は不可欠である。

2.2.1.3 条件分岐

if 構文は「○○ならば○○せよ、そうでなければ○○せよ」という条件分岐を表します。if 構文は以下のような形式です。

```
if <メンバー名>が{状態名}ならば
    (肯定側の記述)
else
    (否定側の記述)
endif
```

論理記述欄に if 構文を追加すると、「if」「else」「endif」の 3 行が自動的に作られます。肯定側だけでなく否定側も記述するよう心掛けることで、要求を網羅的に記述することができます。
なお、else は、〈メンバー名〉が{状態名でない}の意味です。

（例）

```
if <信号機の色>が{青}ならば
    (車を進行せよ)
else
    (車を停止せよ)
endif
```

LD-02 if 構文を使用するときは、肯定側と否定側の両方を記述するよう心掛けることで、要求を網羅的に記述することができる。

論理記述欄に if 構文を追加すると、助詞（「を」や「が」など）を自動的に挿入できます（「環境設定」にて設定）。直接入力することも可です。この助詞の挿入領域は、メンバー名と状態名の間の入力エリアです。if 構文は、一般的な日本語文法でいうと、メンバー名は目的語ないし主語、状態名は述語に相当します。なお、例における{青}以降の「ならば」のような助動詞についても候補リストから選択入力できますし、直接入力も可能です（「環境設定」にて設定）。

LD-03 if 構文の助詞（メンバー名と状態名の間）には「を」や「が」が設定できる。

LD-04 if 構文では、メンバー名を目的語ないし主語、状態名を述語にする。

2.2.1.4 命令形または終止形

Do 構文は「○○せよ」という命令形か、「○○である」という終止形に用います。Do 構文は以下のような形式です。

```
Do <メンバー名>を{状態名}せよ
（例）
Do <車>を{停止}せよ
Do <車>は{停止中}である
```

LD-05 Do 構文は「○○せよ」という命令形を用いる。

LD-06 Do 構文を「○○である」という終止形を用いる。

論理記述欄に Do 構文を追加すると、中置助詞に「を」、後置助詞に「せよ」がそれぞれ自動的に挿入されます。ここで、中置助詞が書かれる領域は、メンバー名と状態名の間の入力エリアです。また、後置助詞が書かれる領域は、状態名の後の入力エリアです。Do 構文の中置助詞と後置助詞はそれぞれ

「を」と「せよ」にすることを推奨しますが、必要に応じて変更しても構いません。これは、日本語文法の用語でいうと、メンバー名が目的語、状態名が述語に相当します。

LD-07 Do 構文の中置助詞（メンバー名と状態名の間）は「を」にする。

LD-08 Do 構文の後置助詞（状態名の後）は「せよ」にする。

LD-09 Do 構文では、メンバー名を目的語、状態名を述語にする。

2.2.1.5 機能を分割する

Fn 構文は機能を複数の単位機能に分割する役割を持ちます。Fn 構文は以下のような形式です。

```
Fn [単位機能名]
```

単位機能が大きい場合（また論理記述でたくさんの構文が記述された場合）、ひとまとまりを下位の単位機能として書くことができます。この場合に、Fn 構文を使います。親子の関係ができます。

この親子関係は概念の分類（分割）と上下の配置、すなわち階層関係を示します。階層関係を明確にすることで、概念の統合と整合性が成立します。

LD-10 論理記述欄の記述が長くなったとき、Fn 構文を使うことで、記述の一部を下位の単位機能に移すことができる。

2.2.1.6 複数の選択肢を持つ条件分岐

switch 構文は複数の選択肢を持つ条件分岐を表します。switch 構文は以下のような形式です。

```
switch <メンバー名>が
case {状態名 1}
    (メンバー名が状態名 1 である場合の記述)
case {状態名 2}
    (メンバー名が状態名 2 である場合の記述)
case {状態名 3}
    (メンバー名が状態名 3 である場合の記述)
elsw
    (メンバー名が状態名 1 でない、かつ、メンバー名が状態名 2 でない、かつ、メンバー名が状態名 3 でない場合の記述)
endsw
```

論理記述欄に switch 構文を追加すると、「switch」、3 行の「case」、「elsw」、「endsw」の 6 行が自動的に作られます。1 個の switch 構文につき 2 個以上の case 構文を入力できます。
switch 構文のメンバー名と状態名は、それぞれ、if 構文のメンバー名と状態名と同じように入力します。
また、switch 構文のメンバー名の後の入力エリアは助詞が書かれる領域、case 構文の状態名の後の入力エリアは助動詞が書かれる領域です。これらはそれぞれ、if 構文の助詞（メンバー名と状態名の間）と助動詞（状態名の後）と同じように入力します。

2.2.1.7 ブロックに何も記述しないことを明示する

SLP の「構文検査」では、ブロックに何も構文が書かれていないと警告を発します。ここで言うブロックとは、if と else の間、else と endif の間、case と次の case の間、最後の case と elsw の間、elsw と endsw の間のことです。

ブロックに何も構文が書かれていないことが間違いではないことを明示するためには、ブロックを Do nothing などの表現の構文で埋めてください。表現は環境設定で変更可能です。

2.2.1.8 コメント

「コメント」構文を使うと、SLP 文法では記述しにくい事柄を自由に記述することができます。

2.2.1.9 改行

「改行」構文を使うと、論理記述欄の意味の区切りを視覚的に表すことができます。

2.2.1.10 ループ

loop 構文、for 構文、while 構文はループを表します。論理記述欄に loop 構文を追加すると、「loop」「endlp」の 2 行が自動的に作られます。
論理記述欄に for 構文を追加すると、「for」「endfor」の 2 行が自動的に作られます。
論理記述欄に while 構文を追加すると、「while」「endwhile」の 2 行が自動的に作られます。
また、exitlp 構文はループを抜けることを表します。ループの構文（ loop, endlp, for, endfor, while, endwhile, exitlp ）は、単にループを表現する「コメント」構文です。
ループの回数の記述は必要なく、後述するようにループの回数に依存した検査はありません。
for 構文、while 構文は以下のような形式です。通常、for 構文のメンバー名 1 とメンバー名 2 とメンバー名 3 は、ループの条件を記載するため、同じメンバー名になります。

for <メンバー名 1>を{状態名 1}せよ; <メンバー名 2>が{状態名 2}; <メンバー名 3>を{状態名 3}せよ
(メンバー名 2 が状態名 2 である場合の記述)
※メンバー名 2 が状態名 2 のとき、上述の処理を繰り返す。初期設定として、メンバー名 1 を状態名 1 にし、繰り返しのたびにメンバー名 3 を状態名 3 にする。
endfor

while <メンバー名>が{状態名}
(メンバー名が状態名である場合の記述)
※メンバー名が状態名のとき、上述の処理を繰り返す。
endwhile

2.2.1.11 用語の命名規則

2.2.1.11.1 メンバー名

メンバー名には、文を構成する対象とその属性のうちの、対象を書きます。この対象は、日本語の文法上では目的語や主語に相当します。これをメンバー名と呼びます。
メンバー名は、if 構文、Do 構文、switch 構文に含まれる入力エリアに書かれます。
メンバー名は、if 構文では目的語または主語、Do 構文では目的語の役割を担います。そのため、メンバー名は名詞や名詞句、名詞節で命名します。
メンバー名は文書全体で一意な名前を付けるのが原則です。そのために、必要に応じて修飾語を付加して、長さにこだわらず最も適切なメンバー名を付けることを推奨します。

NM-01 メンバー名は名詞（名詞句、名詞節でもよい）にする。

NM-02 メンバー名は文書全体で一意な名前にする。

NM-03 メンバー名は長さにこだわらず、一意な名前になるまで修飾語を加える。

メンバー名の命名は、対象の持つ属性との関係に依存します。例えば、信号機は、その色とその作動状況について各々論じる場合には、メンバー名の命名が異なります。簡単な図示をすれば、以下となります。

メンバー名	状態名
信号機の色	赤
	黄
	青

メンバー名	状態名
信号機の作動状況	正常
	故障
	停電

NM-04 メンバー名の命名は状態名との関係で決める。

メンバー名には論理学の記号である量化記号を含めることができます。例えば、「すべての本」「ある本」などのメンバー名は、「すべての」と「ある」が量化記号と認識されます。
量化記号は「量化記号」ウィンドウで設定できます。
「通常文に変換」「条件の論理性と用語の確認（論理式決定表作成）」などの操作で否定文（else または elsw）を出力するとき、量化記号を含むメンバー名は、量化記号を対記号（否定語）に置き換えて出力されます。「すべての本」の否定は「ある本」となります。
なお、量化記号を含むメンバー名として、例えば「すべての男と女」というように複数の概念を含むように記述すると問題が起きます。「すべての男と女」の否定は「ある男と女」として出力されますが、これは正しくありません。正確には「すべての男と女が○○である」の否定は、強いて言えば、「ある男が○○でない、または、ある女が○○でない」です。このような問題を防ぐには、「すべての男と女」のようなメンバー名を使わず、「すべての男」と「すべての女」のように複数のメンバー名に分けることを推奨します。

NM-05 量化記号を含むメンバー名を使うときは、「すべての男と女」のように書かず、「すべての男」と「すべての女」に分ける。

2.2.1.11.2 状態名

状態を名詞として記述できる場合は、状態名は名詞（名詞句、名詞節でもよい）で命名します。
そうでないときは、状態名を動詞で命名します。
ただし、その動詞の活用語尾の扱いは、動詞がサ行変格活用であるか、if 構文の場合であるか、Do 構文の場合であるかによって異なります。
まず、動詞がサ行変格活用であるときは、語幹までを状態名にします。
このとき、if 構文の場合であれば、助動詞に「した」「している」などの活用語尾を入力します。
他方、Do 構文の場合であれば、後置助詞に命令形の活用語尾「せよ」、または「すること」を入力します。
さらに、動詞がサ行変格活用でないときは、if 構文と Do 構文とで状態名の命名方法が異なります。
if 構文では「0 より大きい」などのように終止形で状態名を記述します。一方、Do 構文では、「閉じよ」などの命令文や「閉じる」（こと）などの終止形を記述します。

状態名の品詞	ー		状態名	助動詞/後置助詞	例
名詞、名詞節	ー		名詞、名詞節	である	{本}である
動詞	活用の違い	構文の違い	状態名	助動詞	例
	サ変活用	if 構文	語幹	活用語尾あり	{到達}した
		Do 構文	語幹	すること/せよ	{表示}すること/せよ
	サ変活用でない	if 構文	終止形	無し	{0 より大きい}
		Do 構文	終止形／命令形	無し/こと	{閉じる}こと/{閉じよ}

（例）動詞がサ行変格活用であるとき（「到達する」、「表示する」）

```
if <回答メッセージ>が{到達}している
    Do <回答メッセージ>を{表示}せよ
else
    Do nothing
endif
```

NM-06 「状態を名詞」として記述できるならば、状態名を名詞（ないし名詞句、名詞節）にする。

NM-07 状態名を名詞で表現できないときは、状態名を動詞にする。ただし、その動詞がサ行変格活用であるときは、語幹までを状態名にして、活用語尾を省く。

NM-08 状態名を名詞またはサ行変格活用の動詞で表現できないとき、if 構文では、「0 より大きい」などのように終止形で状態名を記述する。

NM-09 状態名を名詞またはサ行変格活用の動詞で表現できないとき、Do 構文では、「閉じよ」または「閉じる」（こと）などの命令形や終止形で状態名を記述する。

2.2.1.11.3 時制表現

現在や過去などの時制の違いを表現する場合には、通常は、状態名の後ろにある助動詞の入力エリアを利用し記述します。
ただし、このエリアは、コメント的機能ですので、記述の検査の対象にはなりません。あくまでもユーザーが時制を注意する場合に使います。例えば、「たったいま変化した」ことが注意を喚起すべきときなど、このエリアを利用します。
しかし、この時制の違いを条件として生かしたい場合には、時制による違いを状態名に記します。
時制の区分は、次の３種が考えられます。信号の色に例えて述べます。

①	状態変化（現在の状態の変化）	「信号が青になった」
②	現在状態	「信号が青である」
③	過去状態	「信号が青であった」

コメントとして、時制を表現する場合には、①<信号>が{青}になった、②<信号>が{青}である、③<信号>が{青}であった、となります。
他方、時制を条件として生かす場合には、①<信号>が{青になった}、②<信号>が{青である}、③<信号>が{青であった}、となります。

2.2.1.11.4 語句（メンバー名と状態名）をどこまで、どう書けばよいか

SLP では記述された用語の検査を行います。
文字列のチェックしか行いませんので、同じメンバー名は同じ文字で書く必要があります。長い名詞句も同様です。このような場合、「候補リスト」からのメニュー（論理記述の際）を使うと、既存の語句を再利用できます。
状態名の場合も同様に文字列のチェックとなります。先のように時制の違いを要件の違いとして反映したい場合には、「青となった」と「青である」のように書き分ける必要があります。

2.2.1.11.5 単位機能名

単位機能名は、体言止めではなく、動詞を用いることも、事象の動きを表現でき一案かもしれません。
単位機能名は文書全体で一意である必要があります。そのため、必要に応じて修飾語を付加して、十分に長い名前を付けることを推奨します。特に、親単位機能の名前に条件を表す修飾語を加えて子単位機能を命名すると、単位機能名を容易に命名できます。例えば、親単位機能が「○○せよ」であるとき、子単位機能は「××のとき○○せよ」と命名します。最終的には、インデント目次を見ただけで文書全体の流れが分かるように単位機能名が付くようになります。
注意すべきは、既存の要求仕様書の章や節をそのまま単位機能名にする場合です。既存の要求仕様書は必ずしも、機能の階層化を図ったものではないためです。このような場合には、新たな階層（目次）を

作らざるを得なくなります。その場合に、目次名が既存の要求仕様書と同じものではかえって混乱することとなりますので、異なる命名をしてください。
なお、このように既存の要求仕様書を元に SLP 文書を作る場合には、「原要求項目識別子」（単位機能ヘッダー欄）を用いて、オリジナルの要求仕様書とのトレーサビリティを図ることができます。

NM-10 単位機能名は動詞で事象の動きを表現することも一案である。

NM-11 従来の要求仕様書の階層構造にとらわれず、独自の単位機能名を付ける。

NM-12 単位機能名は文書全体で一意にする。

NM-13 「原要求項目識別子」（単位機能ヘッダー欄）を用いて、オリジナルの要求仕様書とのトレーサビリティを図る。

2.2.1.11.6 助詞と助動詞についての補足

助詞と助動詞は論理記述欄の構文を、自然言語ライクに読んでもらうための用途が第一です。
検査の対象になっているわけではありません。
助詞はメンバー名と状態名の間、助動詞は状態名の後にあります。
事前に環境設定で助詞や助動詞に用語を設定できます。

構文を記述する場合に、これらの事前の設定どおりに、助詞が設定されます。
助動詞は if 構文の場合、状態名の後の後置助詞の入力エリアで、右クリックすることで、「候補リストから選択」メニューから、時制の異なる助動詞を選択することができます。また、メニューから候補を選択するのではなく、助動詞を直接入力することも可能です。

2.2.2 要求を自由に記述する

SLP 文法に従って論理記述欄を記述するのは、要求を論理的かつ簡潔に書くためには有益です。しかし、従来の要求仕様書に慣れている人にとって、新しい方法論になじむまでの抵抗感も大きいかもしれません。SLP の特性をより手っ取り早く理解するには、論理記述欄ではなく機能内容欄を主体に使う方法もあります。

GN-01 SLP をより手軽に使うには、論理記述欄ではなく機能内容欄を主体に使っていく。

機能内容欄ではリッチエディットコントロールが有効であり、ユーザーが文字を自由に入力できます。また、フォント（フォントファミリー、サイズ、色、太字、斜体、下線）も変更することができ、他のアプリケーションから、画像、図形、表などをコピー・アンド・ペーストすることができます。また、OLE オブジェクトを挿入することもできます。
既存の要求仕様書と同じように、機能内容欄に自由に書いて、それをインデント目次または階層目次を使って階層的に整理することができます。また、OLE オブジェクトをファイルへのリンクにすることで、複数のファイルに分散した要求仕様書を、SLP 文書を中心にして階層的に管理することができます。

GN-02 既存の要求仕様書と同じように、機能内容欄に自由に書いて、それをインデント目次または階層目次を使って階層的に整理することができる。

GN-03 OLE オブジェクトをファイルへのリンクにすることで、複数のファイルに分散した要求仕様書を、SLP 文書を中心にして階層的に管理することができる。

2.2.2.1 単位機能の作成方法について

単位機能の作成方法には、以下の２つのモードがあります。

(1) 同期モード（【要求仕様化フェーズ】に対応）
このモードでは、単位機能は目次欄と論理記述欄のいずれか一方に作るだけで自動的に両欄に作られます。このケースを「同期」モードと呼びます。
(1.1)　ただし、この同期モードにおいても、論理記述欄で単位機能を（最初に）作った場合には（Fn 構文）、その単位機能は目次欄にも作られます。項番も適切に自動採番されます。これを「確定」的と呼びます。
(1.2)　他方、目次欄で単位機能を（最初に）作る場合には、その単位機能は論理記述欄にも自動的に記述されますが、その位置（行）が適切とは限りません。したがって、これを「暫定」的と呼びます。
「暫定」を「確定」にするには、ユーザーの論理記述欄での修正操作が必要となります。
（Fn 構文を、ブロック内の適切な位置に配置することです。）
(2) 半同期モード　（【要求獲得・分析フェーズ】に対応）
ユーザーは単位機能を目次欄のみで作成する場合があります。この場合には、この単位機能が論理記述欄に自動的に書き込まれることはありません。このケースを「半同期」モードと呼びます。
またこの場合の単位機能を「未定」単位機能と呼びます。

2.2.2.2 単位機能の同名と再利用

プログラムでは同名のラベルは同じ内容を指しますが、要求仕様書の場合はそうではありません。
仕様書の目次には、「概要」、「目的」、「対象」、「用語説明」、「機能概要」、「機能一覧」、「インタフェース」、「非機能要件」など、馴染みのタイトルがありますが、文書により内容が異なります。
本ツールではこのような実際の文書の特性を生かすために、機能を拡張し、単位機能の「同名」を扱えるようになっています。「同名」でも内容は異なる訳です。いわば「同名異体」です。
異なる単位機能を「個別」単位機能と呼びます。「同名」でも「個別」が存在するという訳です。
また、同名同内容（「同名同体」）のものを他の箇所で利用（引用）することもあります。
この機能を「再利用」と呼びます。単位機能としては「共通」と呼びます。
再利用される「共通」には再利用の元となるオリジナルなものと、そうでないものとに分類できます。
これを「正」、「副」とします。
これらの単位機能の違いを理解することは重要です。以下に整理した表を掲げます。

	単位機能の種類	同名	内容	状態
共通単位機能	正	○	○	同名同体
共通単位機能	副	○	○	同名同体
個別単位機能	×	○	×	同名異体
個別単位機能	×	×	―	異名

2.2.2.3 機能内容欄の書き方規則

機能内容欄には要求を自然文で書きます。自由な文体で書くことができます。ただし、いくつかの書き方のルールを守るだけで、SLP 構文の自動作成が効率的に進み、手直しが少なくなります。

2.2.2.3.1 条件文には「格」を入れること

文の中に条件を意味する接続詞があっても、主語や目的語（主格、目的格。以下、格）のない文は、条件文になりません。条件文にするには、少なくとも結論部（後件）には「格」を付けてください。

例えば、格のない文「論理関係文節または属性文節であれば、単位文候補文節である。」は、自動構文化によって、以下のような論理記述になります。

Do<>{論理関係文節または属性文節であれば、単位文候補文節}である。

このような場合、後件に主語を付けると「論理関係文節または属性文節であれば、それは単位文候補文節である。」という記述は、自動構文化によって、以下のような条件文になります。

if <>が{論理関係文節または属性文節である}ならば、
　　Do<それ>は{単位文候補文節}である。
else
　　Do nothing
endif

もちろん、前提（前件）にも格を付けると、前件も後件も格と述語を持った文（単位文）となります。

AS-01 入力の自然文を論理記述でも条件文にしたい場合には、少なくとも入力文の結論部に主語かも目的語を付け、結論部が文（単位文）になるようにする。

2.2.2.3.2 「場合、」の書き方

「～場合、」という条件文の前提になる文が長い複合名詞（名詞の集まりによる名詞句）を持っている場合などは、条件文とならないことがあります。
（例）
単位文候補文節が論理関係文節かつ属性文節である場合、文節の前件後置語が、この単位文の後置語”。

Do<単位文候補文節が論理関係文節かつ属性文節である場合、文節の前件後置語>が、{この単位文の後置語}である。

このような場合には、「場合には」とか、「場合に」などのように助詞を加えると条件文となります。「、」はあってもなくても構いません。その結果は以下です。
（例）
単位文候補文節が論理関係文節かつ属性文節である場合には、文節の前件後置語が、この単位文の後置語である。

if <単位文候補文節>が{論理関係文節かつ属性文節}である場合には、
　　Do<文節の前件後置語>が、{この単位文の後置語}である。
else
　　Do nothing
endif

AS-02 「場合、」には「場合には」や「場合に」などの助詞を付けると、文は確実に条件文となる。

2.2.2.3.3 「のは、、ため」の常套句の書き方

「～のは…ためである」は理由を書く常套文ですが、「の」ではなく、「～理由は…ためである」と書くことをお勧めします。
（例）
自動運転が必要になるのは、事故死０を目指すためである。

Do<自動運転が必要になるの>は、{事故死０を目指すため}である。

「の」で終わる＜～の＞が名詞句となるのは「の」が構文解析器で「名詞」と判定されるからです。これは正しい判定ですが、「の」を名詞句の最後とする用語は（通常は）ないので、「の」を「理由」など、通常の名詞と置換することをお勧めします。

AS-03 「～のは…ためである」の文では、「の」に変え「理由」などとする。

2.2.2.3.4 「例えば、」の位置は文の先頭に

「例えば、」のような、通常は文の先頭に来る常套句は、文の間には書かないようにします。
（例）
「必須」命令は、例えば、「こと。」を削除する。

(:「必須」命令は、
(:例えば、　　（注）“(:”はコメントであることを示す。
Do<>{「。」を削除する}。

そこで、「例えば、」を先頭にすると、以下となります。
（例）
例えば、「必須」命令は「こと。」を削除する。

(:例えば、
Do<「必須」命令>は{「。」を削除する}。

ただし、「例えば、」の後ろの語の対象は、いくつかの候補の中から１つ取り出す、という意味を含んでいます。ただ、自動構文化機能には、ここまで意味を推し量ることはできません。ただし、「少なくとも１つある」という意味を表現する場合には、存在（ヨ、在る）を手動で構文表現できるようになっています。

以上より、当ツールでは「例えば、」を通常は文の先頭に来る常套句として扱っています。類似の表現は、「一例をあげれば、」とか、「以下の例では、」などです。

AS-04 「例えば、」のような、常套句は、文頭に書く。

2.2.2.3.5 括弧の続きは「、」で区切る

括弧が続くと、自動構文化の結果に意味不明なアルファベットが現れる場合があります。

例えば、文「これらの命令のうち、「必須」「任意」「自由」「行頭」「行末」は入力のパターンを形成する。」は、以下のようになります。

Do<入力のパターン>を{これらの命令のうち、ldqjoalnhvhqmnub「自由」「行頭」「行末」は形成する}。

これを避けるには、「これらの命令のうち、「必須」、「任意」、「自由」、「行頭」、「行末」は入力のパターンを形成する。」のように、連続した括弧を読点で区切るようにしてください。

AS-05 括弧が続くときは、読点などで区切る。

2.2.2.3.6 自動構文化処理でコメント行を作る場合

自動構文化機能では、入力が文でない場合は、出力は常にコメント行となり、メンバー名と状態名による文（単位文）は作られません。この機能を利用し、コメント行をつくることができます。入力を文と

して出力させないためには、入力の最後に数字や記号など任意の文字を書きます。これで、その行はコメント行となります。

2.3 SLP 文書を編集する

SLP 文書の編集は、SLP 文書の作成とほぼ同じ操作です。論理記述欄のメンバー名、状態名、助詞、助動詞を変更するには、変更したい入力エリアにカーソルを移動して、文字を削除または挿入します。また、論理記述欄の構文を追加したり削除したりするには、論理記述欄を右クリックしてメニューを選択します。

2.4 SLP 文書の版を管理する

SLP 文書は、1 文書内で 10 版まで、版の管理をすることができます。
一般的な「メジャーバージョン番号＋マイナーバージョン番号」の型式で、版数をファイル保存時に自動更新することもできますし、手入力で明示的な版数を指定することができます。
また更新履歴を表示させることも可能ですので、作業の経緯を振り返ることもできます。

2.5 SLP 文書間の差分を表示する

SLP 文書の改版における差分を表示させ、視覚的に知ることができます。比較は前章の機能で付与された版数を基準に行い、結果を「検査・検索等結果」ウィンドウに表示します。
これにより、改版時のより具体的な修正項目を把握することができます。

2.6 SLP 文書間の単位機能を結合する

SLP 文書間で単位機能を相互にコピーすることができます。
単位機能をコピーして 1 つの文書にまとめるための機能であり、グループでの作業を支援します。
また、派生開発などで、ある単位機能をそのまま移植したいようなときにも使用します。
単位機能の項番を指定することによりコピー元を決定し、コピー先ではその位置を指定します。
目次に追加しないで単位機能一覧にのみ追加させて、後の編集操作で属性を決定することも可能です。

2.7 SLP 文書を分析する

SLP 文書を作成したら、今度はその SLP 文書を分析します。分析の結果をもとに SLP 文書を修正して、より厳密で分かりやすくすることができるためです。SLP 文書を分析する方法には、狭義の検査（構文検査、矛盾検査、冗長検査）、用語に関する機能（曖昧語検索、類似語検索、語句（メンバー名と状態名）のエクスポート）、その他の機能（条件の論理性と用語の確認、条件順序性確認）があります。

2.7.1 SLP 文書を検査する

SLP 文書を検査するには、メインメニューの「検査」以下の項目を使います。
構文検査は、語句（メンバー名と状態名）が空文字列でないかどうかと、ブロックが空でないかどうかを検査します。ブロックを空にすることが正しいと確信でき、構文検査で警告を出す必要がないならば、ブロックに Do nothing 構文を挿入してください。
矛盾検査は、論理的な矛盾を検査します。「矛盾の可能性があります」という警告が出ますが、この警告を出したくないときは、「無矛盾化設定」を使用します。
なお、「検査・検索等結果」ウィンドウの「矛盾の可能性があります」という警告メッセージをダブルクリックすると、すぐに「メンバー属性無矛盾化設定」ウィンドウが開きます。逆に、「文関係設定」

を使用すると、ユーザーが指定した組み合わせで、明示的に矛盾を検出して、警告メッセージを出力させることができます。矛盾とは逆の不可欠な要件の設定も可能になります。
また、同じ前件に対して異なる後件を記述した文の組み合わせが複数存在した場合には、これが仕様作成者の意図した記述であるかを問う警告を発します。どちらの後件が正しいかわからないからです。

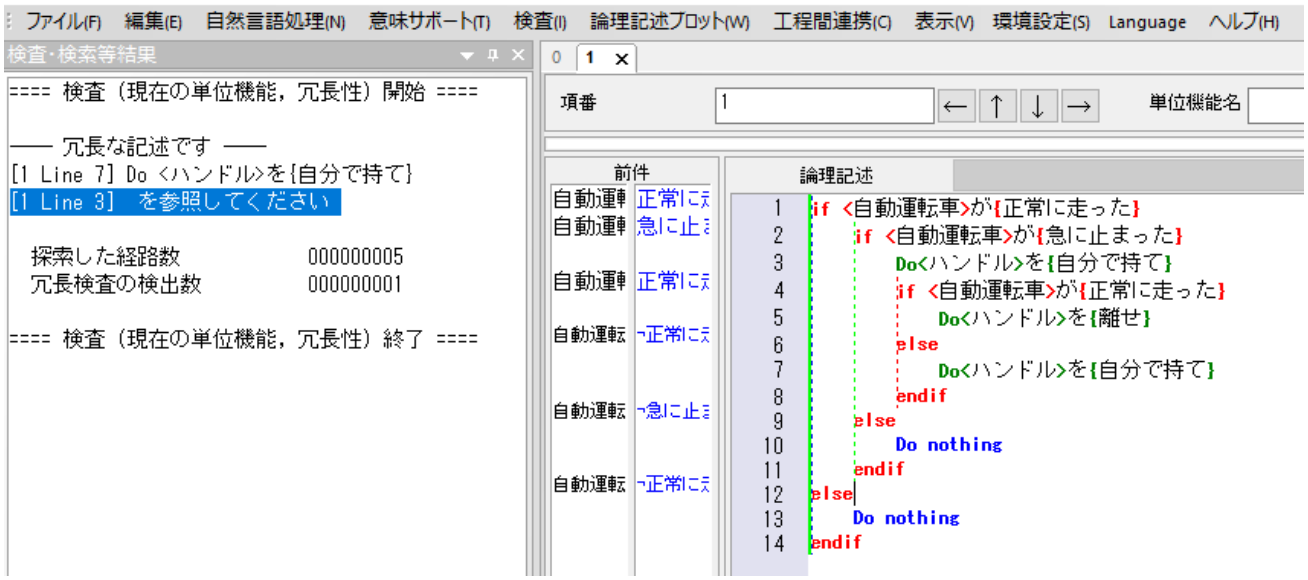
FN-02 矛盾の回避や必要な要件の活用を図るには、「矛盾化設定」と「文関係設定」を用いる。

UI-02 「検査・検索等結果」ウィンドウの「矛盾の可能性があります」という警告メッセージをダブルクリックすると、すぐに「無矛盾化設定」ウィンドウが開く。

冗長検査は、論理的な冗長を検査します。冗長とは端的には文（単位文）の繰り返しを指します。
単位文は語句（メンバー名と状態名）からなりますから、語句を正確に記すことが大事です。SLP の検査は、語句をそれぞれ文字列で比較しているからです。語句の変化や内容までも理解した能力は持っていません。そのため、同じ事柄を意味するメンバー名または状態名は、正確に同じ文字列で記述する必要があります。

NM-14 同じ事柄を意味するメンバー名または状態名は、正確に同じ文字列で記述する。

下記のような論理記述は「冗長な記述です」という検査結果が出ます。



上の図の例を説明すると、自動運転が正常に戻った場合には、ハンドルから手を離せるのですが、事例では正常に戻らなかったので、ハンドルを持ったままです。にもかかわらず、またハンドルを持て、なったものですから、これは冗長とツールは判断した、いうことになります。

次の図は、冗長の指摘を除いたものです。自動運転が正常に戻らなかったので、前の状態のまま（ハンドルを持ったまま）という意味で、「Do nothing」を入れました。

メンバー名と状態名を除く入力エリア（すなわち、助詞、助動詞、コメントの書かれる領域）は検査の対象とはなりません。これらの入力エリアは、単に読む人にとって読みやすくするためにあります。
なお、loop 構文、for 構文、while 構文を使用した場合、ループの回数に依存した検査は行われないことに留意してください。

NM-15 メンバー名と状態名を除く入力エリア（すなわち、助詞、助動詞、コメントの書かれる領域）は、単に読む人にとって読みやすくするためにある。

GN-04 loop 構文、for 構文、while 構文を使用した場合、ループの回数に依存した検査は行われない。

2.7.2 使用を禁止または制限したい用語を管理する

使用を禁止または制限したい用語を管理するには、そのような用語を「曖昧語」に登録します。曖昧語を登録するには「曖昧語登録・検索」ウィンドウを使用します。「曖昧語登録・検索」ウィンドウでは、曖昧語検索を行うことができます。曖昧語検索は、登録されている曖昧語を、機能内容欄と論理記述欄から検索する機能です。

FN-03 使用を禁止または制限したい用語を管理するには「曖昧語登録・検索」を使用する。

2.7.3 表記が揺れている用語を発見する

表記が揺れている用語を発見するには「類似語検索」を用います。これは、論理記述欄の語句（メンバー名と状態名）のうち、類似度が一定以上である用語の組を発見する機能です。二つの文字列がどの程度に類似しているかの決定が求められる、いわゆる類似度の判定には、レーベンシュタイン距離が用いられています。

FN-04 表記が揺れている用語を発見するには「類似語検索」を用いる。その判定基準はレーベンシュタイン距離である。

2.7.4 用語の一覧を作成する

SLP 文書で使われている用語の一覧を作成するには「語句（メンバー名と状態名）のエクスポート」を用います。メンバー名と状態名の一覧は、厳密には「メンバー名」と「メンバー名と状態名の組」の一覧です。なぜなら、状態名はメンバー名に付随して存在する用語のためです。状態名は状態の名前は同

じであっても、メンバー名に依存した意味を持ちます。つまり、状態名は同じ状態名であってもメンバー名ごとに別の意味を持ちます。
メンバー名と状態名の一覧には、「メンバー名」と「メンバー名と状態名の組」のほかに、それらの用語の使用頻度が出力されます。また、それらの用語が「候補リストの作成」で登録されているかどうかや、「曖昧語」に該当するかどうかも併せて出力（エクスポート）されます。
出力ファイルは CSV 形式ですので Microsoft Excel 等で開いて見ることがきます。
また、このファイルを入力（インポート）することで、ほかの SLP 文書に対してメンバー名と状態名を取り込むことができます。

FN-05 SLP 文書で使われている用語の一覧を作成するには「語句（メンバー名と状態名）のエクスポート」を用いる。

NM-16 状態名はメンバー名に付随して存在する用語である。したがって、状態名は同じ状態名であってもメンバー名ごとに別の意味を持つ。

2.7.5 決定表を作成する

決定表は、SLP 文書の論理記述欄に書かれている、前件（if 構文と switch 構文）と後件（Do 構文）の単位文を、連言や含意などで論理式としてまとめたものです。決定表には論理記述欄の記述のうち語句（メンバー名と状態名）が反映されますが、他の入力エリア（助詞、助動詞、コメントの書かれる領域）は決定表には反映されません。

FN-06 決定表は前件（if 構文と switch 構文）と後件（Do 構文）の単位文を、連言や含意などで論理式としてまとめたものである。

LD-11 決定表には語句（メンバー名と状態名）が反映されるが、他の入力エリアのもの（助詞、助動詞、コメント）は反映されない。

2.8 SLP 文書を印刷する

SLP 文書は「印刷設定」メニューにより、選択された項目（複数可）を印刷することができます。
ただし、機能内容欄ではオブジェクトのリンクと埋込み(OLE)に対応していますが、貼り付けられた画像データは印刷されないことに注意してください。

機能内容欄の画像データを印刷したいときは、ファイルメニューの[他ファイルへの出力]-[機能内容欄のエクスポート]機能を使用して、他のアプリケーションで開いて印刷してください。
（画像データが、正常に印刷されます。）

3 SLP を開発プロセスに取り入れる

3.1 要求仕様書を初めから SLP で書く

要求仕様書を初めから SLP で書くことも可能です。要求仕様書が、存在しないか、存在しても何らかの理由でまったく新たに書き直す必要がある場合には、最初から SLP で要求仕様書を書くことを推奨します。SLP に慣れるまでは、紙またはテキストファイルなどに下書きをしてから SLP 文書を書き始める方がよいかもしれません。慣れてくれば、SLP を用いて、下書きから正式なものまであらゆる文書を作成することが可能になります。

GN-05 既存の要求仕様書が存在しないか、質が悪すぎる場合には、最初から SLP で要求仕様書を書く。

GN-06 SLP に慣れていないうちは、紙またはテキストファイルなどに下書きをしてから SLP 文書を書き始める。

3.2 従来の要求仕様書と SLP を併用する

3.2.1 従来の要求仕様書を階層的に整理する

2.1.3 節「要求を自由に記述する」で既に説明しましたが、単位機能の機能内容欄から既存の要求仕様書にリンクを張ることができます。そのことにより、複数のファイルに分散した要求仕様書を、SLP 文書を中心にして階層的に管理することができます。

3.2.2 従来の要求仕様書をもとに SLP 文書を作成する

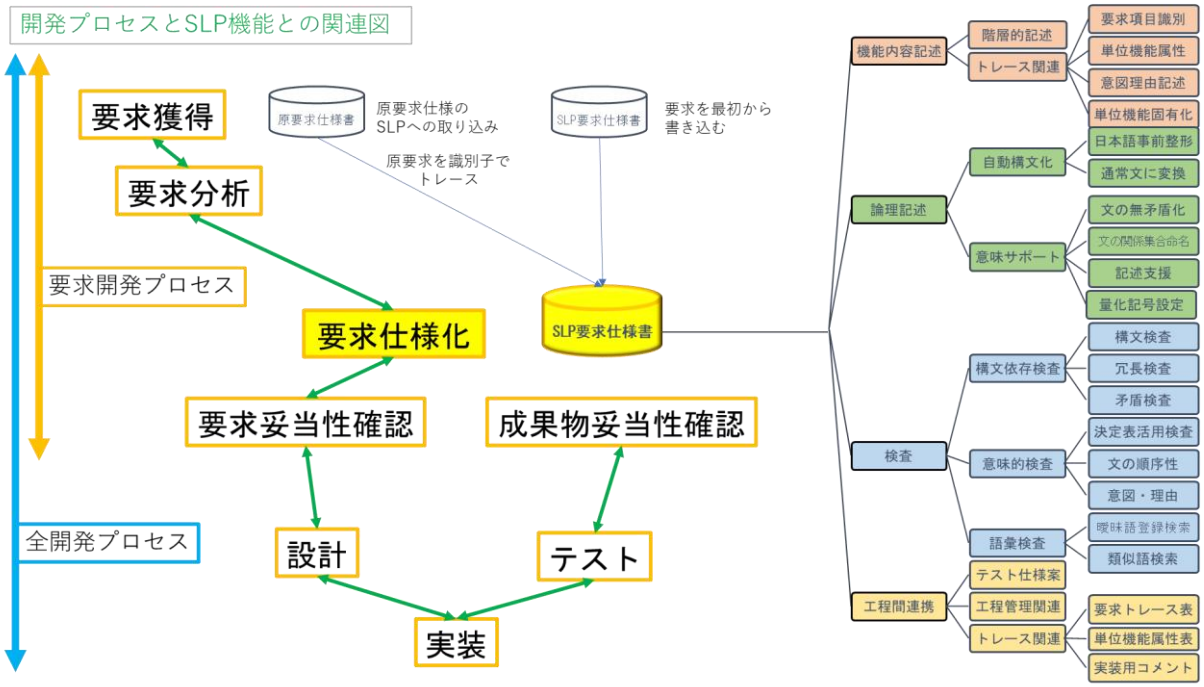
従来の要求仕様書をもとに SLP 文書を作成する方法には、既述のように機能内容欄を使う方法と、論理記述欄を使う方法とがあります。機能内容欄を使う方法とは、2.1.3 節「要求を自由に記述する」で説明したように、機能内容欄を従来の要求仕様書と同じように記述して、それをインデント目次、階層目次または水平目次を使って階層的に整理することです。
一方、論理記述欄を使う方法では、既存の要求仕様書の内容を、SLP 文法に当てはめて記述し直す必要があります。そのためには、以下のような思考過程をとることが推奨されます。

1. 前件（if 構文）と後件（Do 構文）に分割する。
2. 目的語または主語（メンバー名）と、述語（状態名）に分割する。

ここまでの過程で、従来の要求仕様書の記述を SLP 文書の論理記述欄に移すことができます。ここまでを SLP 文書の「下書き」として、さらに続けて、SLP 文書の用語を整理します。
まず、2.1.2 節の命名規則に従って、メンバー名、状態名、単位機能名の用語を整えます。さらに、2.3 節で説明した手法で SLP 文書を分析して、不十分なところがあれば修正します。

LD-12 既存の要求仕様書を SLP の論理記述欄を使って書き直すには、まず前件と後件に分割し、次に目的語または主語と、述語に分割する。続いて、用語の整理と SLP 文書の分析を行う。

以上までの説明を、SLP の機能（メニュー単位）との関連で図示します。



3.3 既存の SLP 文書を修正する

要求仕様書を SLP の論理記述欄を使って記述している場合、既存の要求仕様書を変更または加筆することは、他の形式の要求仕様書（Word、Excel など）よりも容易です。if 構文と Do 構文を用いることで、要求仕様書の構造が明確になっているためです。

GN-07 要求仕様書を SLP の論理記述欄を使って記述しているならば、既存の要求仕様書を変更または加筆することは、他の形式の要求仕様書よりも容易である。

3.4 他の文書(または成果物)と SLP 文書との対応関係を記録する

3.4.1 SLP 文書から原要求を参照する

既存の要求仕様書を SLP 文書に書き換える場合、SLP 文書の記述が要求仕様書のどの記述（要求事項）と関係しているかを記録し、SLP で要求分析など他の工程に進んでいった場合でも、その要件を追跡（トレース）できます。
そのためには、単位機能ヘッダー欄の「原要求項目識別子」テキストボックスと「原要求項目識別子の変更」ウィンドウを用います。
まず、上流の要求仕様書には、項目ごとに何らかのラベルを振っておきます。そして、SLP の側では、単位機能ごとに、その単位機能が、対象の要求仕様書のどの項目を根拠としているか、先ほどのラベルを「原要求項目識別子」に追加することで記録します。
単位機能には複数の原要求項目識別子を記録できます。
また、原要求項目識別子は、単位機能の親子関係に基づいて継承されます。

FN-07 SLP 文書の記述が対象の要求仕様書のどの記述（要求事項）を根拠としているかを記録するには「原要求項目識別子」を用いる。

SLP 文書に記録されている原要求項目識別子は要求トレース表にエクスポートし、一覧することができます（CSV 形式ファイル）。また、逆にインポートすることで原要求項目識別子を変更することができます。もちろんこの場合には、事前に CSV ファイルを変更する必要があります。以下は手順です。

1. SLP で要求トレース表をエクスポートする。
2. Microsoft Excel 等の適切なアプリケーションで要求トレース表を編集する。
3. SLP で要求トレース表をインポートする。

FN-08 原要求項目識別子を一覧するには要求トレース表を用いる。

3.4.2 他の文書に SLP 文書の項目を反映する

前節では、外部の文書（要求仕様書）の項目を SLP 文書に反映させる方法を説明しました（「原要求項目識別子」により）。
ここでは、SLP 文書の項目（単位機能）を他の文書に反映させる方法を説明します。
「ユニークラベル」を使うことでこれを行うことができます。ユニークラベルは単位機能ごとにユニークな識別子（ラベル）を持つからです。（当ツールがユニークなラベルを生成しています。）
「項番」もユニークですが、単位機能の位置を変えると、その単位機能と項番との関係は消えます。これに対して、ユニークラベルは、識別子を変化させようとすると、既存の識別子を削除し、変更履歴に保存し、その上で新たな番号を付けますので、履歴をたどり、単位機能の固有性を追跡することができます。

FN-09 SLP 文書の単位機能に固有のラベルを付けるには「ユニークラベル」を用いる。

3.5 VDM（Vienna Development Method）と併用する

SLP の語句は、VDM（Vienna Development Method）の型の定義に反映させることができます。例えば、メンバー名「信号機の色」が状態名「赤」「黄」「青」を持ち、メンバー名「信号機の動作状況」が状態名「正常」「停電」「故障」を持つとすれば、それを参考に VDM の型の定義を以下のように行うことができます。（自動的に生成するわけではありません。）

```
types
信号機の色型 = <赤> | <黄> | <青>;
信号機の動作状況型 = <正常> | <停電> | <故障>;
信号機型::
        色: 信号機の色型
        動作状況: 信号機の動作状況型;
```

GN-08 SLP の語句は、VDM の型の定義に反映させることができる。

3.6 SLP をテストに使用する

SLP をテストに使用するには「テスト仕様原案作成」を用います。これは、SLP の論理記述欄の記述からテスト仕様書を自動的に作成する機能です。

SLP 文書は論理記述欄で条件の網羅的記述を行っていますから、それがそのままテストの条件表になります。

そのため、当ツールの方法は、テスト技法のひとつである CPM（コピー&テスト&モディファイ）法が利用しやすくなっています。
ただし、この「コピー&テスト」の仕方ですと、要求仕様書に関して、テストの専門家から見た場合の視点を入れることができません。そのようなことから別途仕様書を興すことになります。

しかし、そのように別途仕様書を興す場合にも、SLP を利用することでテスト仕様を作ることを効率化できます。SLP で仕様書を興し、そのまま「テスト仕様原案作成」を実行すれば、CPM ではない、従来からのテスト専門家の視点からのテスト仕様書を作ることができます。

3.6.1 テスト仕様原案作成時の注意事項

テスト仕様原案は「確定」の単位機能を対象として作成されます。「確定」単位機能とは、論理記述欄に Fn 構文として書かれている単位機能を指します。
インデント目次には、■で示されている単位機能です。

FN-10 SLP をテストに使用するには「テスト仕様原案作成」を用いる。

GN-09 SLP から自動生成されテスト仕様書を実際にテストできるように、要求の作成者とテストの実施者との間でテスト方法を事前に共有しておくことが望ましい。

4 アドバイス一覧

4.1 名前の付け方

No.	アドバイス内容
NM-01	メンバー名は名詞（名詞句、名詞節でもよい）にする。
NM-02	メンバー名は文書全体で一意な名前にする。
NM-03	メンバー名は長さにこだわらず、一意な名前になるまで修飾語を加える。
NM-04	メンバー名の命名は状態名との関係で決める。
NM-05	量化記号を含むメンバー名を使うときは、「すべての男と女」のように書かず、「すべての男」と「すべての女」に分ける。
NM-06	「状態を名詞」として記述できるならば、状態名を名詞（ないし名詞句、名詞節）にする。
NM-07	状態名を名詞で表現できないときは、状態名を動詞にする。ただし、その動詞がサ行変格活用であるときは、語幹までを状態名にして、活用語尾を省く。
NM-08	状態名を名詞またはサ行変格活用の動詞で表現できないとき、if 構文では、「0 より大きい」などのように終止形で状態名を記述する。
NM-09	状態名を名詞またはサ行変格活用の動詞で表現できないとき、Do 構文では、「閉じよ」または「閉じる」（こと）などの命令形や終止形で状態名を記述する。
NM-10	単位機能名は動詞で事象の動きを表現することも一案である。
NM-11	従来の要求仕様書の階層構造にとらわれず、独自の単位機能名を付ける。
NM-12	単位機能名は文書全体で一意にする。
NM-13	「原要求項目識別子」（単位機能ヘッダー欄）を用いて、オリジナルの要求仕様書とのトレーサビリティを図る。
NM-14	同じ事柄を意味するメンバー名または状態名は、正確に同じ文字列で記述する。
NM-15	メンバー名と状態名を除く入力エリア（すなわち、助詞、助動詞、コメントの書かれる領域）は、単に読む人にとって読みやすくするためにある。
NM-16	状態名はメンバー名に付随して存在する用語である。したがって、状態名は同じ状態名であってもメンバー名ごとに別の意味を持つ。

4.2 論理記述欄を記述する方法

No.	アドバイス内容
LD-01	「if」「Do」「Fn」の 3 種類の構文は不可欠である。
LD-02	if 構文を使用するときは、肯定側と否定側の両方を記述するよう心掛けることで、要求を網羅的に記述することができる。
LD-03	if 構文の助詞（メンバー名と状態名の間）には「を」や「が」が設定できる。
LD-04	if 構文では、メンバー名を目的語ないし主語、状態名を述語にする。
LD-05	Do 構文は「○○せよ」という命令形を用いる。
LD-06	Do 構文を「○○である」という終止形を用いる。
LD-07	Do 構文の中置助詞（メンバー名と状態名の間）は「を」にする。
LD-08	Do 構文の後置助詞（状態名の後）は「せよ」にする。
LD-09	Do 構文では、メンバー名を目的語、状態名を述語にする。
LD-10	論理記述欄の記述が長くなったとき、Fn 構文を使うことで、記述の一部を下位の単位機能に移すことができる。

LD-11	決定表には語句（メンバー名と状態名）が反映されるが、他の入力エリアのもの（助詞、助動詞、コメント）は反映されない。
LD-12	既存の要求仕様書を SLP の論理記述欄を使って書き直すには、まず前件と後件に分割し、次に目的語または主語と、述語に分割する。続いて、用語の整理と SLP 文書の分析を行う。

4.3 SLP の機能の紹介

No.	アドバイス内容
FN-01	SLP 文法で要求を記述するには論理記述欄を用いる。
FN-02	矛盾の回避や必要な要件の活用を図るには、「矛盾化設定」と「文関係設定」を用いる。
FN-03	使用を禁止または制限したい用語を管理するには「曖昧語登録・検索」を使用する。
FN-04	表記が揺れている用語を発見するには「類似語検索」を用いる。その判定基準はレーベンシュタイン距離である。
FN-05	SLP 文書で使われている用語の一覧を作成するには「語句（メンバー名と状態名）のエクスポート」を用いる。
FN-06	決定表は前件（if 構文と switch 構文）と後件（Do 構文）の単位文を、連言や含意などで論理式としてまとめたものである。
FN-07	SLP 文書の記述が対象の要求仕様書のどの記述（要求事項）を根拠としているかを記録するには「原要求項目識別子」を用いる。
FN-08	原要求項目識別子を一覧するには要求トレース表を用いる。
FN-09	SLP 文書の単位機能に固有のラベルを付けるには「ユニークラベル」を用いる。
FN-10	SLP をテストに使用するには「テスト仕様原案作成」を用いる。

4.4 全般的な注意事項

No.	アドバイス内容
GN-01	SLP をより手軽に使うには、論理記述欄ではなく機能内容欄を主体に使っていく。
GN-02	既存の要求仕様書と同じように、機能内容欄に自由に書いて、それをインデント目次または階層目次を使って階層的に整理することができる。
GN-03	OLE オブジェクトをファイルへのリンクにすることで、複数のファイルに分散した要求仕様書を、SLP 文書を中心にして階層的に管理することができる。
GN-04	loop 構文、for 構文、while 構文を使用した場合、ループの回数に依存した検査は行われない。
GN-05	既存の要求仕様書が存在しないか、質が悪すぎる場合には、最初から SLP で要求仕様書を書く。
GN-06	SLP に慣れていないうちは、紙またはテキストファイルなどに下書きをしてから SLP 文書を書き始める。
GN-07	要求仕様書を SLP の論理記述欄を使って記述しているならば、既存の要求仕様書を変更または加筆することは、他の形式の要求仕様書よりも容易である。
GN-08	SLP の語句は、VDM の型の定義に反映させることができる。
GN-09	SLP から自動生成されテスト仕様書を実際にテストできるように、要求の作成者とテストの実施者との間でテスト方法を事前に共有しておくことが望ましい。

4.5 操作方法

No.	アドバイス内容
UI-01	論理記述欄に構文を追加するには右クリックメニューを用いる。
UI-02	「検査・検索等結果」ウィンドウの「矛盾の可能性があります」という警告メッセージをダブルクリックすると、すぐに「無矛盾化設定」ウィンドウが開く。

4.6 自動構文化

No.	アドバイス内容
AS-01	入力の自然文を論理記述でも条件文にしたい場合には、少なくとも入力文の結論部に主語かも目的語を付け、結論部が文（単位文）になるようにする。
AS-02	「場合、」には「場合には」や「場合に」などの助詞を付けると、文は確実に条件文となる。
AS-03	「～のは…ためである」の文では、「の」に変え「理由」などとする。
AS-04	「例えば、」のような、常套句は、文頭に書く。
AS-05	括弧が続くときは、読点などで区切る。