

## FDT を用いた通信システムの 仕様記述教育支援システム (2) — 仕様記述の演習 —

杉浦茂樹 程子学 白鳥則郎  
東北大学 工学部

近年、厳密で曖昧性のない FDT の必要性が高まっているが、学習性や理解性に難があるため、その普及は十分ではなく、ソフトウェア生産性の向上のためには、FDT に精通していない人のための教育支援環境の提供が急務である。本稿では、FDT を用いた通信システムの仕様記述支援システムの中の演習を支援するサブシステムについて検討する。具体的には、本学 4 年生を対象とした、LOTOS を用いた仕様記述演習の学生実験の結果の分析と、それに基づいた、正規化-LOTOS と LTS を用いた支援システムの設計について述べ、最後に、支援システムの具体的な構成を与える。

## **A Education Support System for Specification of a Communication System based on FDT (II) — Exercise of Specification Description —**

Shigeki SUGIURA, Zi-xue CHENG, Norio SHIRATORI  
Faculty of Engineering, TOHOKU University

The expert FDTs user, especially a user for LOTOS, is very few and as a consequence the use of FDTs is no wide spread, in spite of its wide applicability. To advance the productivity of a protocol design, we need to provide a user-friendly environment for protocol development to help beginners of FDT who consist of greater part of users community. For this reason, we develop a support system for specification based on FDT. We first discuss the results of student experiment for specification description exercises using LOTOS. Then the design of a support system is discussed, and finally detail design of the support system is also given.

## 1. まえがき

近年の情報通信システムの大規模化、複雑化によって、厳密で曖昧性のない形式記述技法 (FDT) の必要性が高まっている。しかし、FDT は従来の非形式的な記述技法に比べて、学習性や理解性に難があると言われており、その普及は十分ではない。ソフトウェア生産性の向上のためには、FDT に精通していない人のための教育支援環境の提供が急務となっている。

本稿では、本学 4 年生に対して行っている学生実験の一つである「情報通信ネットワーク」を対象とした、FDT を用いた通信システムの仕様記述教育支援システム [3] の中の演習を支援するサブシステムについて検討する。具体的には、FDT の一つである LOTOS を用いた仕様記述演習の実験結果の分析と、それに基づいた、正規化-LOTOS と LTS を用いた支援システムの設計について述べ、最後に、支援システムの具体的な構成を与える。正規化-LOTOS と LTS を導入することによって、従来の方法より一層広いタイプの解答に対して支援を行うことを可能とした。

## 2. 仕様記述環境

### 2.1 SEGL (Specification Environment for G-LOTOS)[1]

SEGL は LOTOS による仕様記述、とりわけグラフィカル表現である G-LOTOS を支援するために設計、試作された環境である。SEGL には標準的なエディタの機能に加えて、構文誤りを防ぐための機構が取り入れられている。SEGL は、図 1 に示すように、(1) G-LOTOS エディタ、(2) T-LOTOS エディタ、(3)  $G \rightarrow T$  変換機構、(4)  $T \rightarrow G$  変換機構の 4 つのモジュールによって構成されている。

本学生実験で、仕様記述は G-LOTOS エディタを用いて行うが、解答結果の解析などは、 $G \rightarrow T$  変換機構により T-LOTOS に変換したものを元に行う。

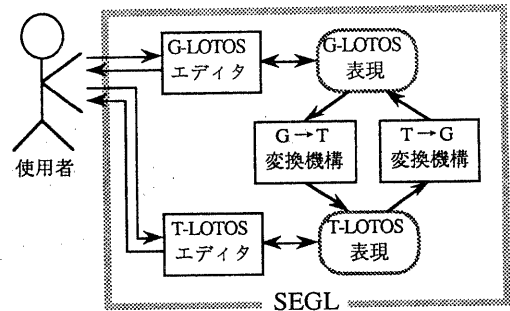


図 1 SEGL の構成

### 2.2 MBP (Model Based Programming environment)[2]

MBP は仕様記述の支援環境であり、具体的には LOTOS を対象としている。MBP では、使用者から対話的に要求を獲得して、その要求から自動的に形式仕様を生成する機能を提供することによって、FDT の文法や記述スタイルを強く意識することなく仕様記述を行うことができる。MBP は、図 2 に示すように、(1) 要求獲得機構、(2) 知識モデル、(3) 変換機構、(4) チュータの 4 つのモジュールによって構成されている。

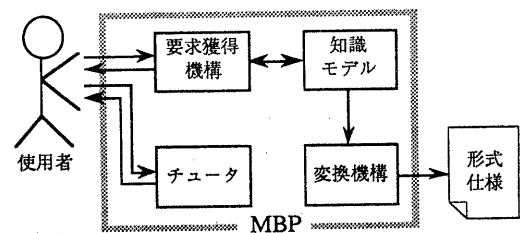


図 2 MBP の構成

## 3. 演習問題の分析

### 3.1 演習問題

#### (1) 課題 1: 自動販売機①

##### a) 仕様の日本語表現

110 円のジュースの自動販売機を LOTOS で仕様記述せよ。

110 円の入れ方は、

- 1) 100 円玉を入れてから 10 円玉を入れるか、
- 2) 10 円玉を入れてから、100 円玉を入れるの 2 通りしかできないものとする。

b) 模範解答

```
specification vending_machine1[c10,c100,juice] :noexit
behaviour
  (c100; c10; exit [] c10; c100; exit)
  >> juice; vending_machine1[c10,c100]
endspec
```

c) 正解の LTS 表現

課題 2 の正解の LTS 表現を図 3 に示す。

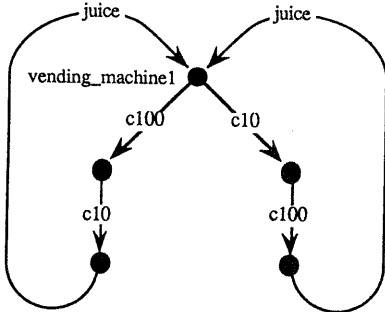


図 3 課題 1 の正解の LTS 表現

(2) 課題 2: 早押しクイズスイッチシステム

a) 仕様の日本語表現

図 1 において、解答者 A (スイッチ A) と B (スイッチ B) のうち、先にスイッチを押した人がクイズに答える権利をもつ。

スイッチ A, B を押すと、それぞれ赤, 黄のランプが点灯する。ただし、両方のスイッチが同時に押されても唯一つのランプしかつかないものとする。

スイッチ A, B および赤いランプと黄色いランプに関する早押しクイズのスイッチ=ランプシステム (図 1 の点線で囲んだ部分) の動作を LOTOS で記述せよ。

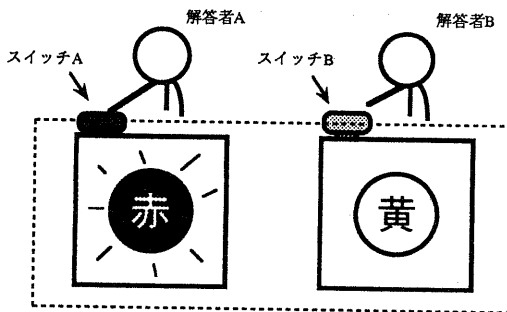


図 1. 早押しクイズシステム

b) 模範解答

```
specification quiz[a,b,red,yellow] :noexit
behaviour
  (a; red; exit [] b; yellow; exit)
  >> quiz[a,b,red,yellow]
endspec
```

c) 正解の LTS 表現

課題 2 の正解の LTS 表現を図 4 に示す。

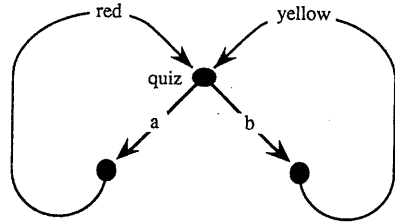


図 4 課題 2 の正解の LTS 表現

(3) 課題 3: 扇風機

a) 仕様の日本語表現

図 2 のように、扇風機に「切」、「弱」、「強」の 3 つのスイッチがある。「切」の停止状態において「弱」(または「強」) を押すと「弱風」(または「強風」) の状態となる。

図 3 は「弱」が押された状態を示す。この状態では「切」または「強」を押すことによって「切」か「強風」の状態に移すことができる。

このような状態の切り替えは繰り返し何回でもできる。

ただし、同じスイッチを 2 回、3 回と連続しては押さないものとする。(同じスイッチを続けて押しても意味はない。)

以上に述べた扇風機の機能を LOTOS で記述せよ。

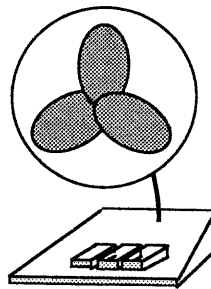


図 2

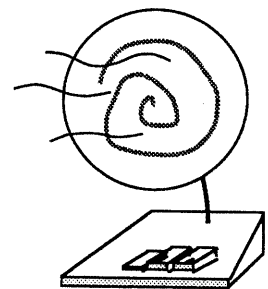


図 3

b) 模範解答

```

specification fan[off,weak,strong] :noexit
behaviour
  fan_off[off,weak,strong]
where
  process fan_off[off,weak,strong] :noexit :=
    weak; fan_weak[off,weak,strong]
    []
    strong; fan_strong[off,weak,strong]
where
  process fan_weak[off,weak,strong] :noexit :=
    off; fan_off[off,weak,strong]
    []
    strong; fan_strong[off,weak,strong]
endproc
process fan_strong[off,weak,strong] :noexit :=
  off; fan_off[off,weak,strong]
  []
  weak; fan_weak[off,weak,strong]
endproc
endproc
endproc

```

c) 正解の LTS 表現

課題 3 の正解の LTS 表現を図 5 に示す。

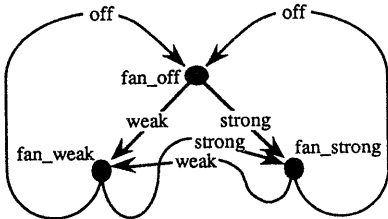


図 5 課題 3 の正解の LTS 表現

(4) 課題4: 自動販売機②

a) 仕様の日本語表現

次のような機能を持つ自動販売機を LOTOS で仕様記述せよ。

販売機は 120 を受け取ったところでコーラ一本を提供する。

120 円の入れ方は

- 1) 100 円玉 1 枚と 10 円玉 2 枚の組み合わせ
- 2) 50 円玉 2 枚と 10 円玉 2 枚の組み合わせ

のみが許されるものとする。

b) 模範解答

```

specification vending_machine2[c10,c50,c100,cola]
:noexit
behaviour
  ((c100; exit ||| c10; c10; exit)
  []
  (c50; c50; exit ||| c10; c10; exit))
  >> cola; vending_machine2[c10,c50,c100,cola]
endspec

```

c) 正解の LTS 表現

課題 4 の正解の LTS 表現を図 6 に示す。

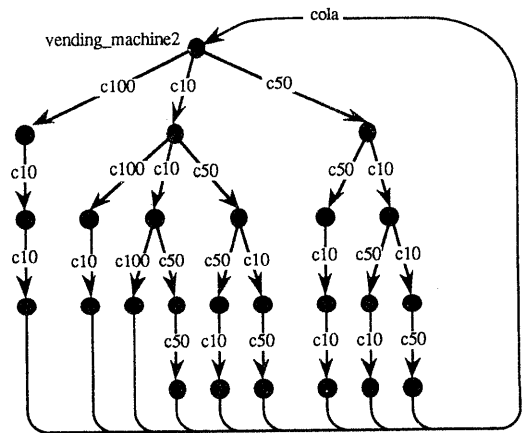


図 6 課題 4 の正解の LTS 表現

3.2 誤りの分類

学生の解答の誤りを次の観点から分類した。

(1) 構文誤り

- a) 識別子の構成に関する誤り
- b) 識別子リストの構成に関する誤り

(2) 静的な意味誤り

- a) ファンクショナリティに関する誤り
- b) プロセス・スコープに関する誤り

(3) 動的な意味誤り

- a) 構文要素の知識の欠落
- b) 仕様の記述を行うための知識の欠落

(4) その他

- a) 不完全な解答

表1 解答の構文誤りに関する分析結果

構文誤りの種類	課題1	課題2	課題3	課題4
プロセス識別子の構成に関する誤り	4/8(人)	0/8(人)	0/9(人)	3/8(人)
ゲート識別子の構成に関する誤り	7/8	0/8	0/9	5/8
ゲート・リストの構成に関する誤り	1/8	0/8	0/9	0/8

表2 解答の静的意味誤りに関する分析結果

静的意味誤りの種類	課題1	課題2	課題3	課題4
ファンクショナリティに関する誤り	5/8(人)	2/8(人)	1/9(人)	5/8(人)
プロセス・スコープに関する誤り	0/8	1/8	1/9	0/8

表3 解答の動的意味誤りに関する分析結果

動的意味誤りの種類		課題1	課題2	課題3	課題4	
正解	模範的な解答	6/8(人)	3/8(人)	0/9(人)	2/8(人)	
	冗長な解答	1/8	0/8	0/9	2/8	
誤り	構文要素の知識の欠落	選択を使用すべきところに非同期的並列を用いた	0/8	1/8	0/9	0/8
		選択を使用すべきところに割り込みを用いた	0/8	1/8	0/9	0/8
	仕様の記述を行うための知識の欠落	非決定性に関する理解が不十分	0/8	1/8	0/9	0/8
		状態の表し方が分かっていない	0/8	0/8	5/9	0/8
	不完全解答		1/8	0/8	1/9	1/8
	無解答		0/8	2/8	3/9	3/8

注: 実験を受けた学生の総数は8人である。課題1、課題2、課題4においては一学生一解答であるが、課題3のみ複数解答した学生が一人いるため、学生総数は9人となっている。

### 3.3 解答結果とその分析

本節では、3.2 節の分類に基づいて、解答を解析した結果について述べる。

解答の構文誤りに関する分析の結果を表 1 に、静的意味誤りに関する分析の結果を表 2 に、動的意味誤りに関する分析の結果を表 3 に示す。

## 4. 演習支援の基本方針

3 章での演習問題の分析結果から、演習の支援方法について検討する。

### 4.1 正規化-LOTOS と LTS に基づく支援

本支援システムでは、従来からの学生が記述した LOTOS そのものを用いた支援のみではなく、それを処理して得られる、正規化-LOTOS と LTS を用いた支援も行なう。

#### (1) 正規化-LOTOS

正規化-LOTOS とは、LOTOS に対して、構文要素の種類などの情報量を減少させずに、冗長さのみを減らすようなルールを適用して変換を行ったものである。この変換ルールには、例えば、

rule1)  $a; B1 \square a; B2 \rightarrow a; (B1 \square B2)$   
(共通アクション・プレフィックスの括り出し)

rule2)  $B \text{ exit } \gg \text{ stop } \rightarrow B \text{ stop}$   
(不要な順次合成の削除)

などがある。

本支援システムでは、専門知識中に持っている、正解の知識、および、誤りの知識と、学生の解答との比較を正規化-LOTOS によって行ない、それに基づいて支援を行っている。

#### (2) LTS

一般に、LOTOS を用いて仕様を記述したときには、使用する構文要素の選択の自由度などによって、さまざまな書き方を行なうことができる。しかし、これらを LTS に変換したときには、すべて、唯一の LTS へと変換される。

これに着目して、本支援システムでは、専門知識中に持っている正規化-LOTOS による知識と学生の解答が一致しなかった場合には、LTS を用いた支援を行う。

### 4.2 支援する誤り

本支援システムで、支援する誤りは、大きく分けて (1) 構文誤り、(2) 意味誤りの 2 種類である。このうち、(2) 意味誤りについては、さらに、① 静的意味誤り ② 動的意味誤りの 2 種類に分けられる。

(1) 構文誤りとは、仕様の記述が LOTOS の構文規則に反している場合である。これには、識別子の構成の誤りなどがある。

(2) 意味誤りとは、仕様の記述が LOTOS の構文的には正しいが、その動作が要求通りには働かない場合である。

① 静的意味誤りとは、要求の動作と記述された仕様の動作を比較しなくても判断することができる誤りである。これには、ファンクショナルリティの誤りなどがある。

② 動的意味誤りとは、要求の動作と記述された仕様の動作を比較したときのみ判断することができる誤りである。これには、構文要素の知識の欠落による誤りなどがある。

### 4.3 支援の流れ

4.1 節、および、4.2 節の方針に基づいて、本支援システムでは、図 6 のような流れで支援を行う。

まず、学生によって記述された仕様は、構文チェックが行われ誤りの検出が行われる。ここで、誤りが検出されたときには、構文誤りに関する支援を行う。誤りが検出されなかったときには、静的な意味チェックが行われる。ここで、誤りが検出されたときには、静的意味誤りに関する支援を行う。誤りが検出されなかったときには、動的意味誤りに関する支援を行う。

動的意味誤りの支援は、まず、正規化-LOTOS に基づいた支援を試みる。もし、正

正規化-LOTOS に基づく支援ができない場合には、LTS に基づいて支援を行なう。

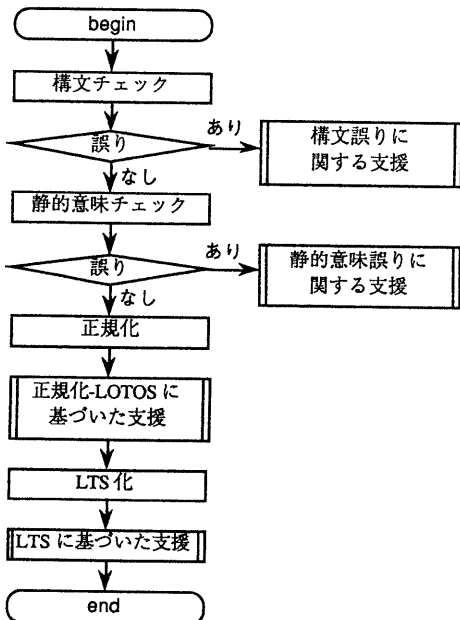


図6 支援の流れ

#### 4.4 支援方法

(1) 従来のコンパイラで用いられてきた技法を利用して行う支援

…入力された LOTOS そのものを用いる  
構文誤りに関する支援、および、静的意味誤りに関する支援は、従来のコンパイラで用いられたきた技法を利用して行う。

(2) 本研究において新しく考案した方法に基づいて行う支援

…入力された LOTOS を処理して得られる、正規化-LOTOS と LTS を用いる  
動的意味誤りに関する支援は、本研究において新しく考案した方法に基づいて行う。

#### 4.5 動的意味誤りに関する支援

動的意味誤りに関する支援は、図7に示される正規化-LOTOS に基づいた支援と、図8に示される LTS に基づいた支援で構成される。

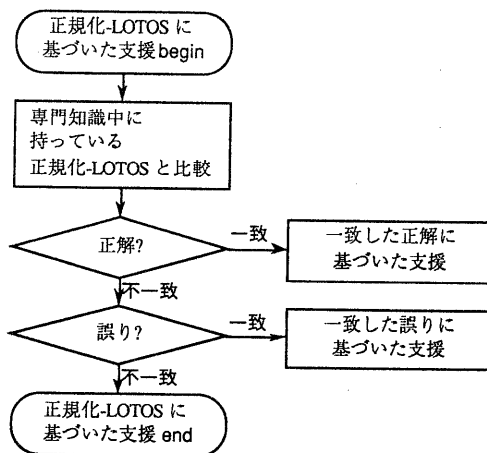


図7 正規化-LOTOS

正規化-LOTOS に基づいた支援は、専門知識中に持っている、正解の知識、および、誤りの知識の正規化-LOTOS と、学生が記述した LOTOS を変換して得られた正規化-LOTOS とを比較して、一致した場合に、それに基づいた支援を行うというものである。

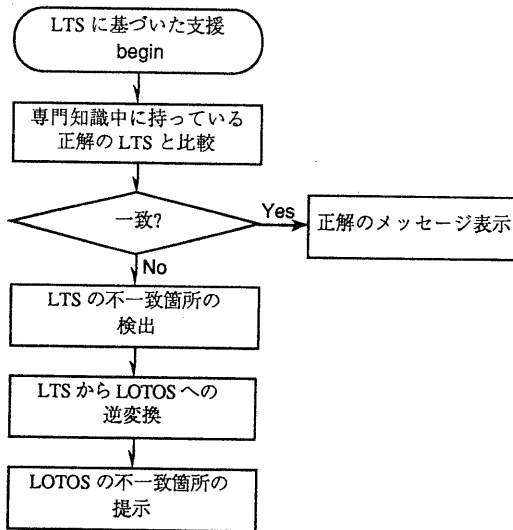


図8 LTS に基づく支援

LTS に基づいた支援は、正規化-LOTOS で支援が行えなかったときに行うもので、正解は LTS に変換すると唯一の LTS で表されることを利用したものである。まず、学生が記述した LOTOS を LTS に変換して、専門知識中に持っている正解の LTS との比

較を行う。もし、一致すれば、正解であることを学習者に示して、支援を終了する。一致しなかった場合には、LTS での比較結果と逆変換機構を用いて、元の LOTOS の誤りの部分を検出し、それを学習者に提示するというものである。

## 5. 演習の支援システム

演習の支援システムは、図9のように、7つのモジュールから構成されている。

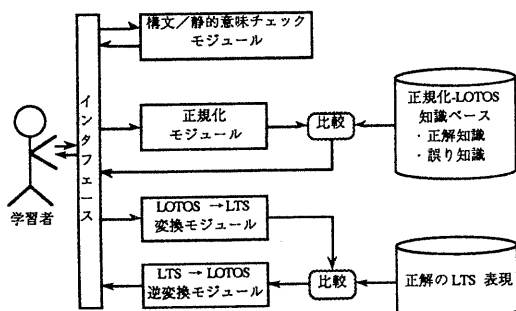


図9 演習支援システムの構成

### (1) インタフェース

学習者とシステムのインタラクションを行うためのモジュール。

### (2) 構文/静的意味チェックモジュール

学習者の記述した LOTOS の構文チェック、および、静的意味チェックを行い結果を出力するモジュール。

### (3) 正規化モジュール

学習者の記述した LOTOS を正規化-LOTOS に変換するモジュール。

### (4) LOTOS → LTS 変換モジュール

学習者の記述した LOTOS を LTS に変換するモジュール。

### (5) LTS → LOTOS 逆変換モジュール

LTS から LOTOS への逆変換を行うモジュール。

### (6) 正規化-LOTOS 知識ベース

課題の正規化-LOTOS による解答を有している。これには、正解の知識のみではなく、誤りの知識も含まれている。

### (7) 正解の LTS 知識

課題の正解の LTS を有している。

## 6. まとめ

今後、より多くの演習結果を解析し、正規化-LOTOS の知識ベースの構築、および、支援システム設計の詳細化を行い、支援システムの試作をして、有効性の確認を行う。

## 参考文献

- [1] 李殷碩, 森健一, 白鳥則郎, 野口正一: "G-LOTOS の仕様化環境 SEGL の構成と試作", 情報処理学会論文誌, Vol.32, No.3, pp.314-323 (March 1991).
- [2] E.S.LEE and N.SHIRATORI: "A Model Based Specification Environment for LOTOS", 日韓 Joint Conf. (SST91-23, SSE91-28, IN91-28, CS91-32), pp.23-28 (July 1991).
- [3] 王家鏞, 程子学, 白鳥則郎: "FDT を用いた通信システムの仕様記述教育支援システム (1) —ネットワークと LOTOS の学習—", 情報処理学会, マルチメディア通信と分散処理研究会技報, (July 1992).