

リアクティブ・データフロー型ドキュメントブラウザにおける  
ユーザレベルへの適応表示制御

7J-4

福田 晴元 高橋 直久

NTT ソフトウェア研究所

1 はじめに

ソフトウェア改造では、ソースコード、設計ドキュメント、テストデータ、ハードウェアの解説文書など多量の技術文書(ドキュメント)から必要な情報を持つドキュメントを検索し、内容を吟味する作業を行う。リアクティブデータフロー型ブラウザ<sup>1)</sup>は、これらの作業の支援をするためドキュメント分析プロセスを記述し再現する機能を提供する。

本稿では、このブラウザに対して、ユーザの違いをレベルの違いとして表現し、各ユーザレベルに応じてドキュメント分析プロセスを再現して表示を行うための実行制御法を与える。また、任意のユーザレベルに適応するドキュメント分析プロセスの記述法について述べる。

2 リアクティブ・データフロー型ブラウザ

リアクティブ・データフロー型ブラウザは、改造に必要な情報を持つドキュメントをノードとし、ドキュメント間の依存関係をアークとしたデータフローグラフ(表示プログラムと呼ぶ)を、データフロー計算モデルの実行規則に従って実行することによりドキュメントを表示する。表示プログラムは、ドキュメントを表示する機能を持つドキュメントノードと実行を制御する制御ノード(手続き呼び出し、手続き開始、手続き復帰、分岐、分配、合流)により記述される。

図1にブラウザの機能構成の概要を示す。実行制御

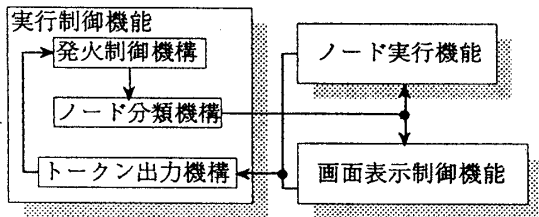


図1: ブラウザの機能構成の概要図

機能は発火制御機構、ノード分類機構、トークン出力機構からなる。発火制御機構は全ての入力アークにトークンが揃ったノードを検出し発火させる。ノード分

Variable-level Browsing for a Reactive Data Flow Document Browser  
Harumoto FUKUDA and Naohisa TAKAHASHI  
NTT Software Laboratories

類機構は、発火したノードがドキュメントノードの場合は画面表示制御機能へ、その他のノードの場合はノード実行機能へ制御を移行する。トークン出力機構は出力アークへトークンを出力する。ノード実行機能はノードに与えられた機能を実行する。画面表示制御機能は発火したドキュメントノードに対応するドキュメントを画面上に表示する。以上の機能により表示プログラムを実行した場合には、図2に表すようにドキュ

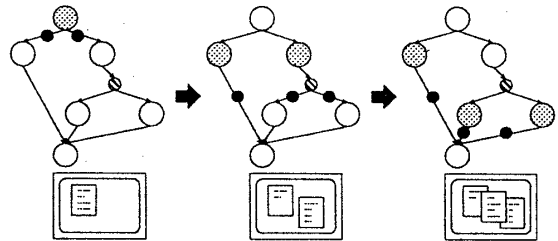


図2: 実行例

メントが表示される。図において小さい黒丸はトークン、斜線のノードは分配ノード、白丸はドキュメントノード、網掛けの丸はドキュメント表示中のドキュメントノードを表す。四角は画面イメージを表す。

3 レベルによるユーザのモデル化

以下のようにユーザに対して「レベル」の概念を導入することによりユーザの違いをモデル化する。

- レベルの高いユーザほど少ないドキュメントで情報を取得する。
- レベルの低いユーザはレベルの高いユーザが必要とするドキュメントに加えて、更に多くのドキュメントから情報を取得する。

以上のようにレベルの高いユーザが必要とするドキュメントの集合と、レベルの低いユーザが必要とするドキュメントの集合は包含関係にあると定める。

ここで、ユーザに対してユーザレベル  $l_i (i = 1, 2, \dots, n)$  を与える。  $j < k$  のとき  $l_j > l_k$  とし、  $l_j$  は  $l_k$  よりレベルが高いという。ユーザレベル  $l_m$  の開発者が必要とするドキュメントの中でユーザレベル  $l_{m-1}$  以上の開発者は必要としないドキュメントをレベル  $l_m$  のドキュメントと呼ぶ。すなわちこのモデルでは、ユーザレベル  $l_m$  のユーザはレベル  $l_0 (l_0 \geq l_m)$  のドキュメント全てが必要であるとみなされる。

表 1: ノード  $n$  の環境レベルの満たすべき条件

ノード種別	入力アーク先ノードに起因する条件	出力アーク先ノードに起因する条件
ドキュメント, 演算, 手続き開始, 分配	$level(n) \leq level_{in}(n, i)$ なる $i$ が存在する	$level(n) \leq level_{out}(n, j)$ なる $j$ が存在する
手続き呼び出し, 合流	$level(n) \leq level_{in}(n, 1) = level_{in}(n, 2)$	$level(n) \leq level_{out}(n, 1)$
分岐	$level(n) \leq level_{in}(n, 1) = level_{in}(n, 2)$	$level(n) \leq level_{out}(n, 1) = level_{out}(n, 2)$
手続き復帰	$level(n) \leq level_{in}(n, 2)$	-

4 実行制御法

実行を行うユーザレベルを実行環境レベルと呼ぶ。ある実行環境レベルで実行した際に、そのレベルに必要なドキュメントのみを表示するために、以下のように表示プログラムと実行制御機構を与える。

**実行対象となる表示プログラム** 表示プログラムの全てのノードに対して環境レベルとよぶレベルを付与する。ドキュメントノードには3節に述べた方法に従い環境レベルを設定する。ここで、ノード  $n$  の環境レベルを  $level(n)$  とし、 $n$  の第  $i$  入力アークの接続ノードの環境レベルを  $level_{in}(n, i)$ 、 $n$  の第  $j$  出力アークの接続ノードの環境レベルを  $level_{out}(n, j)$  と表す。

**実行制御機構** 実行環境レベルを  $l$  とする。この時、 $l \leq level_{in}(n, i)$  を満たす環境レベルを持つノードから  $n$  へ張られたアークと、 $n$  から  $l \leq level_{out}(n, j)$  を満たす環境レベルを持つノードへ張られたアークをノード  $n$  のアクティブアーク<sup>2)</sup>と呼ぶ。図1の発火制御機構に対し、入力アークの中で全てのアクティブアーク上にトークンが揃ったときにノードを発火する機構を与える。また、トークン出力機構に対し、出力アークの中で全てのアクティブアークに対してトークンを出力する機構を与える。これにより、実行環境レベル  $l$  で表示プログラムを実行させると  $l \leq level(n)$  の環境レベルを持つノードだけが実行される。

5 レベル付きグラフの作成法

ノードに対して自由にレベルを与えると実行しないノードが発生する可能性がある。また、一度作成されたドキュメント表示プログラムの修正が必要になった場合に、ノードを自由につけ変えていった場合も実行しないノードが発生する可能性がある。図3に例を示す。図において黒丸のノードは分配ノード、白丸のノ

ードはドキュメントノードを表す。ノードの左上の数字はノード番号を表し、右上の数字は環境レベルを表す。(a)は正常な表示プログラムである。(b)はノード5の環境レベルが1であるために、実行環境レベル2で実行した際にノード3以降のノードを実行できない。(c)は(a)にノード9をつけ加えたものである。実行環境レベル3の場合に、ノード9の環境レベルが3であるにもかかわらず実行されない。

このようなレベル付けの誤りにより発生する異常を以下に示す。

- 可到達性異常 ある実行環境レベルで実行を行った際に開始ノードから終了ノードまでトークンが到達するための経路が存在しない。
- 全実行性異常 実行環境レベル  $l$  で実行した際に環境レベル  $level(n) (l \leq level(n))$  をもつ実行されないノードが存在する。

表1に示す条件を満たすように表示プログラムに対して環境レベルの付与または修正をすると、環境レベルを付与する以前の表示プログラムに異常が存在しない場合に、任意の実行環境レベルで実行した際に上述の異常が発生しない表示プログラムを作成できる。

6 おわりに

リアクティブデータフロー型ドキュメントブラウザにおいて、ユーザの違いを「レベル」によりモデル化し、各レベルのユーザに必要なドキュメントだけを表示する機能について述べた。また、可到達性異常、全実行性異常を生じないレベル付き表示プログラムの作成法を示した。今後は任意の実行環境レベルにおける異常の検出法とユーザレベルの設定法を与えるユーザモデルについて検討する。

最後に、日頃御討論頂く伊藤正樹リーダーはじめ、グループの皆様に深謝します。

参考文献

- 1) 福田晴元 高橋直久, リアクティブデータフローモデル: ソフトウェア設計ドキュメントの表示過程の記述と再現, 情報処理学会 ソフトウェア工学研究会, Feb 1993.
- 2) 高橋直久 鈴木英明 直井邦彰 福田晴元, データフロー計算の一般化 - ソフトウェア・リエンジニアリングにおける複雑な情報の構造化と理解を目指して -, 日本ソフトウェア科学会第10回大会, C9-2, Jun 1993.

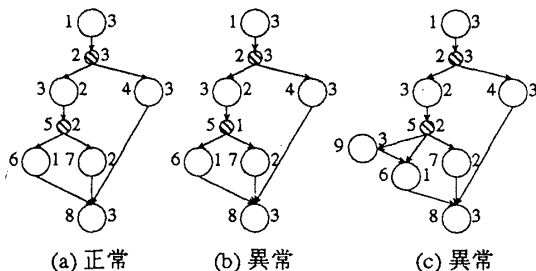


図 3: 環境レベルを付与したプログラム例