

反復的な改善を含むソフトウェアプロセス制御システムの試作と考察

齊藤 良和 橋本 信也

1. 背景と目的

ソフトウェアプロセス管理に関する表現モデルの課題として以下がある。

- (1) 反復的な作業の表現ができること。
- (2) スケジュールの変更に即時に対応できること。
- (3) 曖昧性がなく、視覚的に見やすいこと。

しかしながら、このような課題を解決する開発工程の形式的な表現モデルとして決定的なものは少ない。

本研究では、上記の課題を解決するプロセス記述言語の形式的表現方法を定め、その有用性を確認するためのシステムを試作することを目的とする。

2. 研究内容

既存の表現モデルを Observer パターンの考え方に沿って統一化し、開発プロセスを一般して表現できるようにしたプロセス記述言語 UPFL (Unified Process Flow Language) を考案し、評価した。

3. UPFL の概要

3.1 UPFL の基本構造

UPFL は、図 1 に示すように作業中の状態を示す State-object (以後 S-object) と作業の関連、および作業開始・完了条件を示す Transition-object (以後 T-object) の二種類のノードからなる有向グラフ構造で、監視式によりプロセスの流れを制御するプロセス記述言語である。

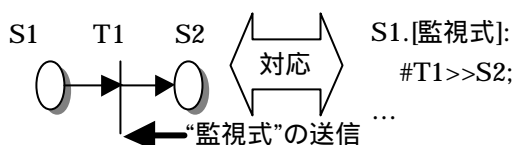


図 1. UPFL グラフと UPFL による構文表現

3.2 UPFL の基本動作

UPFL では、図 2 に示すように Observer パターンのオブジェクト間協調動作の関係を基礎に持つ T-object と S-object とが動的依存関係により、互いの変更を監視し合うことを基本的な動作としている。

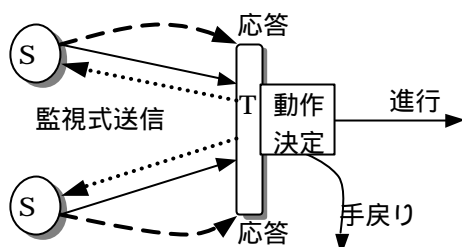


図 2. S-object, T-object 間の協調動作

3.3 UPFL の特徴

UPFL の特徴は以下の通りである。

- (1) 反復作業を含むプロセス管理を表現できる。
- (2) 日程計算や動的な変更によるプロセスの修正にも柔軟に対応できる。
- (3) プロセス制御をグラフ構造で明示的に表現している。

4. 試作システムの機能

前述の特徴を含む UPFL の試作システムの主な機能として以下がある。主な実行画面は図 3 に示す通りである。

- (1) 日程計算
各作業予定時間や、クリティカルパスを求める。
- (2) 監視式での評価結果による条件動作
T-object に入力した監視式から疑似コードを生成し、S-object はその評価結果を返し、T-object 動作決定をする。
- (3) 手戻りによる再作業処理
手戻りが発生した場合、指定した作業まで戻り、プロセス変更や日程再計算を行い、そこから作業を再開する。
- (4) プロセス間の連鎖アポート表現
ある作業が手戻りになるとき、連鎖アポート関係になる作業を表現することができる。

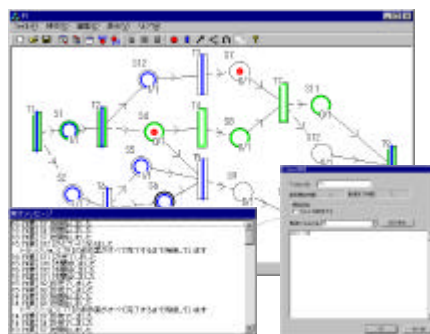


図 3. 試作システムの実行画面

5. 評価とまとめ

試作ツールを用いて、UPFL と他のプロセス表現図式を現実作業で発生する問題について比較評価した結果、表 1 に示す通り 4 つの評価項目に関して有効に表現・制御できることを明らかにした。

表 1. 表現モデルの制御比較

評価項目	Pert 図	PetriNet	UPFL
1.手戻り表現	×		
2.日程計算		×	
3.動的な条件変更	×	×	
4.連鎖アポート制御	×		

これにより、Observer パターンの考え方を拡張し、S-object と T-object に動的依存関係を持たせ、監視しあう動作を持つ UPFL の有用性が確認できた。

今後の展望として、UPFL 構文の洗練と、試作ツールをソフトウェアプロセス管理の支援ツールとして機能拡張していくことが考えられる。

参考文献

- [1]井上克郎, 松本健一, 飯田元: “ソフトウェアプロセス”, 共立出版
- [2]大木幹雄: “動的依存関係を利用した分散オブジェクトの協調機構に関する提案”, 電子情報通信学会 信学技報 KBSE98-4 Vol.98 No.239 (1998)