

PAT Pro 並行システム設計支援ツール

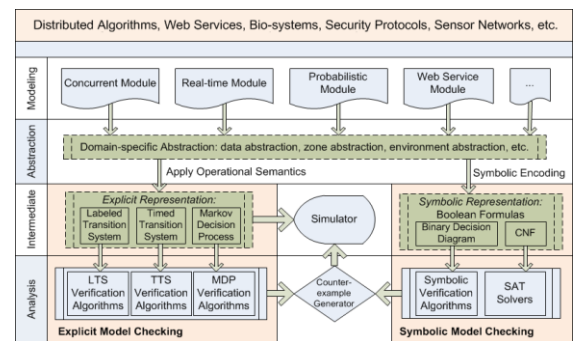
時代はマルチコアへ。今後、ハードウェアを有効利用するためには並列処理をソフトウェアを適切に設計し、実装する必要があります。しかし、並行処理システム(マルチタスク、マルチスレッドを含む)の安全な設計方法が確立されたとはまだ言えないのが現状です。そのようなマルチタスク・マルチスレッドのソフトウェアを設計段階で検査し、妥当性や安全性を確保する必要があります。**PAT Pro**を利用することでそのような設計環境を確保できます。

■ 並行処理記述に適したモデル言語

PAT Pro のモデル記述言語は安全な並列処理設計のためのモデル言語であるCSP(※)をベースにしています。

■ ヴィジュアルな統合環境

モデルエディタ、シミュレータ、モデル検査の機能がGUIによって連携・統合された使いやすいモデル検査環境を提供します。



※ CSP(Communicating Sequential Process)はHoareが提唱した「プロセス代数」モデルです。

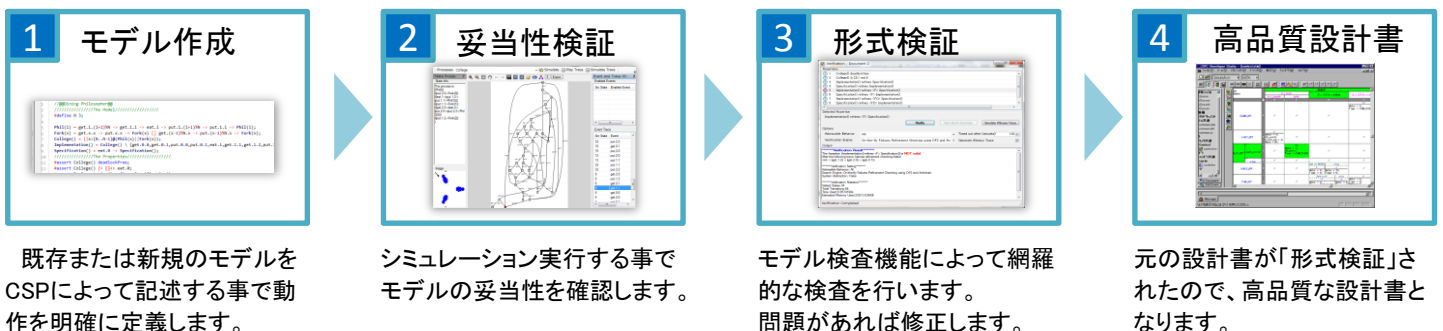
こんな方にオススメ

- ・振る舞い設計を検証し実装時の不具合を減らしたい。
- ・使いやすいモデル検査ツールが欲しい。
- ・用途に合ったモデル言語を選択したい。

期待される効果

- ・設計時のミスが減らすことで**品質が向上**します。
- ・習得時間が短いため、作業**コストを削減**します。
- ・**PAT Pro**では**複数の検査モデル言語が用意されています。**

作業の流れ



関連サービス

- ・お手持ちの設計書から検査モデルを作成支援作業をお手伝いします。
- ・本ツールを利用するような研究を支援します(コンサル、モデル作成、プロトタイプ開発)。
- ・すでに使用中の設計ツールの保存形式を**PAT Pro**のデータへ変換するツールを受託開発します。

PAT Pro PAT Pro の機能紹介

PAT Proのベースとするモデル言語である**CSP**は、並列処理設計手法としての長い歴史を持ちます。複雑になりがちな並行プロセスの設計を**PAT Pro**を利用して検討することで、安全で見通しの良い並列設計を実現可能になります。

使いやすい検査モデルエディタ

PAT Proのモデルエディタは入力ワードの補完機能をはじめとして使い勝手が配慮されています。また、言語要素が色付で表示されるため、モデルの構造が把握しやすくなっています。

```

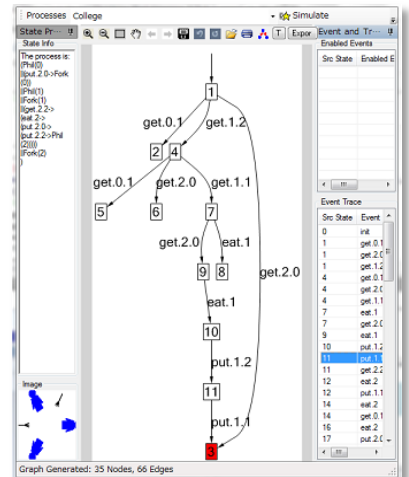
1 //@@Dining Philosopher@@
2 ////////////////////////////////////////////////////////////////////
3 #define N 3;
4
5 Phil(i) = get.i.(1)&N -> get.i.1 -> eat.i -> put.i.(1)&N -> put.i.1 -> Phil(i);
6 Fork(x) = get.x -> put.x.x -> Fork(x) [] get.(x-1)&N.x -> put.(x-1)&N.x -> Fork(x);
7 College() = ||:{0..N-1}@Phil(x)|Fork(x);
8 Implementation() = College() \ (get.0.0,get.0.1,put.0.0,put.0.1,eat.1,get.1.1,get.1.2,put.1.1,put.1.2);
9 Specification() = eat.0 -> Specification();
10 ////////////////////////////////////////////////////////////////////
11 #assert College() deadlockFree;
12 #assert College() != []> eat.0;

```

高機能なシミュレーション機能

PAT Proのシミュレーション機能は、作成したモデルの妥当性をチェックします。また、モデル検査結果としてエラーが検出された場合に、そのエラーへ至るまでのトレース情報が分析可能です。

- ・システム全体のトレース情報の可視化
CSPモデルとして定義された複数の並行プロセスモデルを合成した状態の遷移トレースを動的に生成し、可視化します。
- ・動的なシーケンス図によるトレース
シミュレーションの進行とともにシーケンス図が動的に生成されます。
- ・ステップ毎の変数内容確認
ちょうどビジュアルなデバッガのように、適当に選ばれたトレースの途中ステップでの変数内容を確認できます。



モデル検査機能

- ・反例解析はシミュレーションと連動
モデル検査の結果がエラーだった場合、初期状態からエラーに至るまでのトレースを分析(いわゆる「反例解析」)するために、結果がシミュレーション画面と連動しています。
- ・便利な検査用キーワードの提供
典型的な検査式が書きやすいように、「デッドロックしない」、「到達性」、「発散しない」、「決定的である」など表現するキーワードが用意されています。
- ・「CSP流儀」の検査式の利用
伝統的なCSPを利用した検証項目である、「詳細化関係」を検査できます。これにより、抽象度の異なる2つのプロセスの関係を評価することができます。

