

■ PolySpace Client for C/C++ 7

実行時エラーの存在しないソースコードの証明

PolySpace Client™ for C/C++は、PolySpace Server™ for C/C++と連携し、コードに基づく検証により、オーバーフロー、ゼロ除算、配列の領域外へのアクセスや、その他の実行時エラーがソースコードに存在しないことを証明します。コードの検証には抽象解釈の理論を応用し、プログラムの実行、コードのインストルメンテーション、テストケースを必要とせず、手書きコード、生成コード、または、それらの組み合わせのコードをコンパイルおよびテスト前に検証できます。

PolySpace Client for C/C++の利用

PolySpace Client for C/C++は、ソフトウェアコンポーネントをデスクトップコンピュータ上で検証するための管理機能と表示機能を提供し、ソースコードをファイル単位またはクラス単位で検証処理します。PolySpace Server for C/C++を併用し、検証ジョブをコンピュータクラスタに送信が可能です。

PolySpace Client for C/C++のコマンドライン、グラフィカルユーザインタフェース、またはEclipse™で、以下のことができます。

- ソースコードのインポート
- ターゲット、クロスコンパイラ、またはその他のオプションによるプロジェクトのカスタマイズ
- MISRA-C® やJSF++ (Joint Strike Fighter Air Vehicle C++)のコーディング規約準拠のチェック
- PolySpace Server for C/C++に送信されたジョブの状態の監視
- サーバからデータをダウンロードして検証結果のレポートや表示
- 検証結果の文書化と記録

主な機能

- ソフトウェアコンポーネントのファイルやクラスレベルでの検証
- 抽象解釈を応用
- 実行時エラーをコード内に直接表示
- MISRA-C : 2004およびJSF++コーディング規約違反の検出とソースファイルへのリンク
- EclipseおよびMicrosoft® Visual Studio® IDEとの統合

P
r
o
v
e
n

```
static void Pointer_Arithmetic (void)
{
  int array[100];
  int i, *p = array;

  for(i = 0; i < 100; i++, p++)
    *p = 0;

  if(get_bus_status() > 0) {
    if (get_oil_pressure() > 0)
      *p = 5;
    else
      i++;

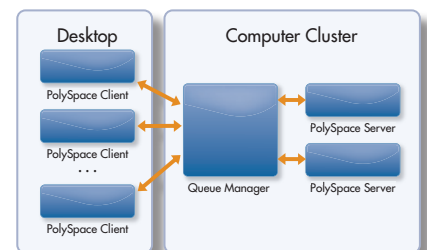
    i = get_bus_status();
    if (i >= 0) { *(p-i) = 10; }

    if ((0 < i) && (i <= 100)) {
      p = p - i;
      *p = 5;
    }
  }
}
```

Green: reliable
Red: faulty
Gray: dead
Orange: unproven

PolySpace Viewerは、検証結果を色分けして表示します。

PolySpace Client for C/C++とPolySpace Server for C/C++によるコード検証ワークフロー。キューマネージャは、PolySpace 検証要求を受信し、ジョブの実行可能な最初のサーバを選択します。



検査対象エラー

オーバーフロー、アンダーフロー、ゼロ除算、その他の算術エラー

配列の領域外へのアクセスと、不正なポインタ参照

未初期化変数の読み取りアクセス

デッドコード

NULL の this ポインタへのアクセス (C++)

オブジェクト指向プログラミング、継承、および例外処理に関連する動的なエラー (C++)

未初期化のクラスメンバ (C++)

その他のエラー (危険な型変換など)

PolySpace Client for C/C++は複数のワークフローをサポートします。たとえば、以下のことが可能です。

- コンピュータクラスタまたはコンピュータファーム上で実行しているPolySpaceサーバに複数の検証ジョブの送信
- 1台のクライアントを使用して、デスクトップコンピュータ上でソースファイルやクラスの検証
- 1台のサーバと複数のクライアントを使用して、複数名の開発者による同時の検証結果の表示や分析

結果の確認

コード内の各要素の状態は以下のように色分けして示されます。

グリーン: 実行時エラーは発生しない

レッド: 演算を実行する度に必ずエラーが発生する

グレー: 到達しない(機能的な問題を示すこともある)

オレンジ: 未確認

MISRA-C : 2004およびJSF++コーディング規約違反の検出も可能です。クライアントによるメトリクスやレポート生成を使用してコードの信頼性と品質の測定と向上を促します。

コード検証結果の分析方法は、開発プロセスの段階、ソフトウェアコンポーネントの重要度、コードの認証要求により選択できます。PolySpace Client for C/C++には定義済みフィルタとカスタマイズ可能なフィルタが用意されており、これらを使用した反復可能な作業として検証結果をレビューします。また、PolySpace Client for C/C++はオレンジとして検出されたコードのテストも可能なため、検証結果をレビューした後、生成されたテストケースを使用して該当箇所のテスト実行が可能です。

PolySpace Client for C/C++による実行時エラーが存在しないことの確認とコードの品質向上の測定により、開発者やテスト担当者、プロジェクトマネージャは、実行時エラーのないコードという目標設定、納品、評価が可能です。

プログラムダイナミクスの検証

従来のバグ検出ツールでは、ソフトウェアのエラーの見落とし(偽陰性)、または必要以上に多くの警告(偽陽性)が発生します。一般的に有限個のテストケースによる動的テストでは、エラーを完全に見逃してしまう場合もあります。PolySpace Client for C/C++は、コードの各箇所の命令毎に変数の取り得る値のすべてを考慮に入れ、プログラムの実行条件をすべて検証します。その検証結果は、コードの各演算の正式な診断結果になります。

PolySpace Client for C/C++により、ソフトウェア開発ワークフローにおける以下のようなすべての重要な作業をサポートします。

- 特定および任意の条件下でのソフトウェアコンポーネントの完全性と品質の検証
- コード品質の傾向のモニタリング
- テスト前のコーディング段階でのエラーの検出と修正
- ソフトウェアコンポーネントに実行時エラーが存在しないことの証明

動作環境

PolySpace Client for C/C++には、以下の製品が必要です。

- PolySpace Server™ for C/C++

動作環境の詳細につきましては、www.mathworks.co.jp をご覧ください。

関連製品

- PolySpace Client™ for Ada
- PolySpace Model Link™ SL (Simulink® 用)
- PolySpace Model Link™ TL (dSPACE® TargetLink® 用)
- PolySpace Server™ for Ada
- PolySpace UML Link™ RH (Telelogic® Rhapsody® 用)