

C++ライブラリの認定：いいとこ取りの関数 vs 完全なヘッダ

C++ library qualification: cherry picking functions versus full headers

2022/10/20



自動車、工業、医療、航空、ロボティクス、鉄道といった業界で多くの企業がクリティカルなC++アプリケーションの安全性を真剣に考えていることを嬉しく思います。カメラやレーダー、他のセンサーからのリアルタイムデータを処理する最新のセーフティクリティカルアプリケーションでは、より高い抽象度を提供し、かつC言語の性能に近い、成熟したプログラミング言語へのニーズがかつてないほど高まっています。C++をフルに活用するには、認定されたC++ライブラリが必要です。しかし、認定はどのようにすればよいのでしょうか？

C++標準ライブラリは巨大で、言語が更新されるたびに増大し続けています。このライブラリを認定するためには、いくつかの重大な設計上の決定や多大な労苦、無数の詳細事項の考慮が必要になります。

一般的なC++のアプリケーションでは、C++ライブラリのコンポーネントすべてが使用されるわけではありません。そこで、顧客から特定のアプリケーションに対するC++ライブラリの認定を依頼された場合、実際にはC++ライブラリの小さなサブセットの認定を検討することになります。しかしこれには、将来アプリケーションを変更する際に、認定されたサブセットのライブラリだけを使用するよう制限される、というリスクがあります。

C標準ライブラリの単一機能の関数の検証は（比較的）簡単です。ほとんどの関数が独立して個別のファイルで定義されているため、どのコードを検証し解析する必要があるのか簡単に分かります。C++の場合、話はまったく異なります。まず、実装の大部分はヘッダファイルに含まれており、関数の宣言ごとに1つのファイルだけが存在するというわけではありません。次に、ヘッダのコードは通常テンプレートベースのものでコンパイル時に評価されるため、カバレッジ解析の難易度は高まります。さらに、C++ライブラリはクラスベースであるので、関数がグループ化され、その実装が絡み合ったものになります。

そのため、単一機能の関数ではなくヘッダごと全てで認定することに、我々は惹きつけられます。テストの要件、内容、実行時のユースケース等へのトレーサビリティが明確になり、認定のプロセスとその結果に関するレポートの透明性にとって非常に有益になるからです。

ヘッダごと全てで認定する我々のアプローチは、顧客から要求される認定済み関数のリストとは異なります。ただ幸いなことに、より透明性の高い認定プロセスに加えて、将来アプリケーションの変更が必要になった場合に、より自由度が高くなるという利点は十分に理解されています。

C++ライブラリ認定のための要件ベースのテストスイートに関する私どもの進捗状況について詳しくお知りになりたい場合、お気軽にお問い合わせください。

Dr. Marcel Beemster, CTO



富士設備工業株式会社 電子機器事業部 www.fuji-setsu.co.jp