C++ユーザのためのパッケージマネージャの開発

— poac: 開発を加速させるパッケージマネージャ —

1. 背景

近年，急速にソフトウェア技術が発展を続けており，それに伴って多くのプログラミング言語が開発されてきた．これらの言語には，汎用性の高い関数などを纏めたライブラリが存在し，言語のインストール時に付随する標準ライブラリと，外部からダウンロードする必要のある外部ライブラリの2種類に分類されることが多い．外部ライブラリを扱うには，外部からダウンロードを行うだけでなく依存関係解決などの作業も必要になるが，このような作業を全て担ってくれるのがパッケージマネージャである．Pythonにはpip，Elixirにはhexといったパッケージマネージャがあるように，C++にもconanというパッケージマネージャがある．しかし，conanは他言語のパッケージマネージャと違い，インタフェースが直感的でない等の理由から，C++ユーザから敬遠されている．

それに加え，アプリケーションをビルドするための使いやすいビルドシステムも存在しない．MakeやCMake等が存在するがCMakeは特に使いづらく，特に文法が異様なためにC++とは別に勉強が必要となる．こういった問題は，初学者からC++が忌避される原因に他ならない．

このままでは，C++にはパッケージマネージャが存在しないのと等しく，また，言語の勉強のために言語のみの勉強に留まらないため，C++は入門しやすい言語とは言えない．パッケージマネージャとビルドシステムは扱う情報の一部が同じであるため，統合する方が相性が良い．そこで，パッケージマネージャ兼ビルドシステムを直感的なインタフェースで提供することで，初学者の入門とパッケージの生成の循環，即ちエコシステムを生み出すことができる．

2. 目的

本プロジェクトでは，初学者でも使いやすいパッケージの整備と，それらを管理するパッケージマネージャを開発することを目的とした．クライアントサイドには，様々なコマンドでパッケージ管理を行うことができるコンソールアプリケーションを提供する．また，それらのコマンドに対応するAPIサーバと，パッケージやドキュメントなどを検索・表示できるWebサーバを提供する．

本プロジェクトは，乱立したライブラリ群を同一のシステム上に纏め，共通のインタフェースで提供することで，C++をより初学者が入門しやすい言語にしていき，また，C++での開発において欠かせないツールとなることを目標としている．

3. 開発の内容

本プロジェクトでは，直感的なインタフェースで使用することが可能なC++のパッケージマネージャ兼ビルドシステム「poac」を開発した．図1にpoacのクライアントサイドのアーキテクチャを示す．クライアントサイドはC++で書かれており，推論機構とコマンド，コマンドが使用するための各種コンポーネントによって実現される．大まかな流れとして，poacバイナリに引数を与えると推論機構でどのコマンドを実行するのか推論し，コマンド部分でコンポーネントを活用しながら各コマンドの機能を構成する．

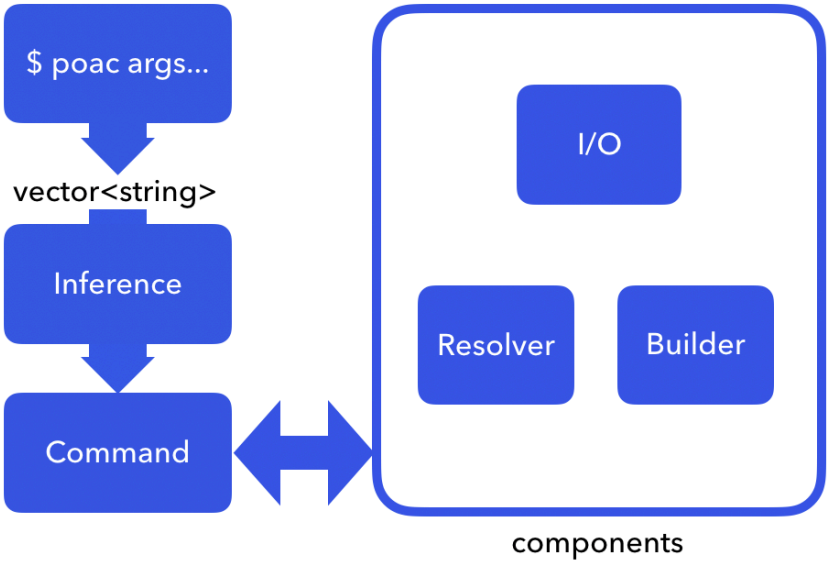

図1. poacのクライアントサイドのアーキテクチャ

- 推論機構

推論機構では，各コマンドへの推論規則をstd::unorderdmapとして用意し，各コマンドの関数ポインタを配列としてコンパイル時に広げておく．それを実行時に一意に指定することで，コマンドへ残りの引数をムーブすることができる．また，規則として存在しないコマンドの指定があればヘルプを表示するようにしている．

- コマンド

コマンドは基本的に肥大化しないように，なるべくコンポーネントを扱うようにしている．そのため，基本的は処理の流れのみが記述されるようになっている．

- I/O

I/Oでは，ファイル操作やパス操作，コマンドラインでの出力を行うためのユーティリティを提供する．

- Resolver

Resolverでは，Semantic Versioning用のライブラリや，基本的なバックトラック型のDPLLアルゴリズムのSATソルバを提供する．それらを用いて依存関係を解決する．

- Builder

  Builderでは，コンパイラの引数を処理するユーティリティや，実際のディレクトリ構造を解析したり，設定ファイルを読むことでビルド設定を作成するユーティリティが実装されている．Builderを使用することでプロジェクトをビルドすることができる．

poacは，`curl -fsSL https://sh.poac.io | bash`コマンドでインストールできる．そのコマンドでインストール可能なOSとアーキテクチャは以下に示す通りで，それ以外はboostとyaml-cppにCMake，C++17がコンパイル可能なコンパイラのインストールが可能である環境であれば，ほぼ全ての環境でビルド・インストールすることができる．

- macOS (>= sierra)
- Linux (= x86_64 GNU/Linux)
- Windows (= Windows Subsystem for Linux)

4. 従来の技術（または機能）との相違

C++にはconanという既存のパッケージマネージャが存在するがインタフェースがモダンでなく，特に初学者にとって使いづらい．しかし，本プロジェクトで開発したpoacは，図2で示すように非常に直感的に使用することができる．

本プロジェクトの特徴はビルドシステムを内蔵していることで，それによりインストールからビルドまでをマネジメントすることが可能となっている．また，一般的な機能に留まらず，依存関係をグラフ画像化するコマンド（図3）や，パッケージの検索を行うコマンド，テストを行うコマンドなども提供している．

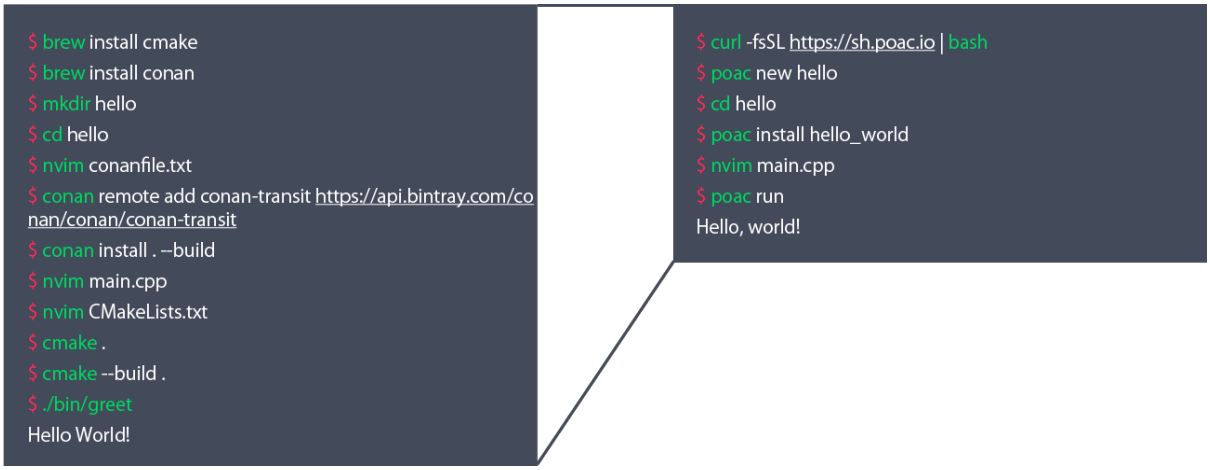

図2: 既存システムとの実行例比較（左：conan，右：poac）

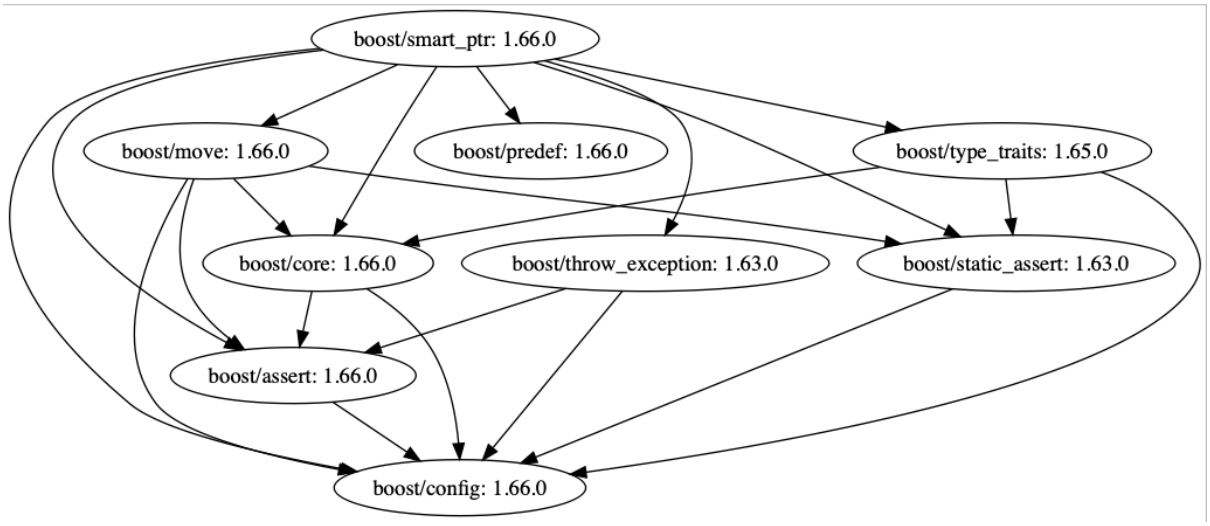

図3: boost/smart_ptrの1.66.0の依存関係

5. 期待される効果

今までC++に入門していなかったユーザが，本プロジェクトを土台にしてC++に入門することが期待される．そこからC++に魅了される者が現れ，より優れたパッケージを公開し，それを新たなC++初学者が使用する，といったエコシステムが構築される．こうした循環の中で，C++によって新たなサービスが生み出されることで，より良い産業の発展が見込める．

6. 普及（または活用）の見通し

本プロジェクトはOSSとして公開されており，多くの貢献者によって機能修正を受けている．今後は，機能・コマンドの追加やバグ修正等を行なっていく．また，そうした中でC++での開発において欠かせないツールとなることを目指す．

7. クリエータ名（所属）

松井 健（立命館大学 情報理工学部 情報理工学科）

（参考）関連URL

- poac: https://poac.io
- Poac document: https://docs.poac.io
- GitHub | poac: https://github.com/poacpm