C++、C++/CLI、C# 適材適所

とっちゃん
επιστημη
えムナウ

高萩 俊行 **Microsoft MVP Windows - SDK**
福田 文紀 **Microsoft MVP Visual C++**
児玉 宏之 **Microsoft MVP Visual C#**

Microsoft Community Ring

おすすめストリートライブ・BoF

8/22（水）16:40 - 17:00 わんくま同盟
「わんくまストリート ライブ ショート ショート」

8/23（木）12:20-12:35 えムナウのプログラミングのページ
「C++・C++/CLI・C# 適材適所のBoFの紹介」

8/23（木）13:30-13:45 わんくま同盟
「わんくまストリートライブ ダイナミック言語のおもしろさ」

8/23（木）15:25-16:40 わんくま同盟
.NET Framework 3.0 をこう使いたい

8/23（木）17:00-18:15 えムナウのプログラミングのページ
C++・C++/CLI・C# 適材適所

アジェンダ

- はじめに
- C++、C++/CLI、C# の比較
  - **生産性**
  - **拡張性**
  - **規模感**
  - **得意分野**
- 3人がこう思う使いどころ
- まとめ

はじめに

- C++ 、 C++/CLI 、 C# は、兄弟のような存在ではあるが、各々が現役で置き換わるような言語ではない。
- επιστημηの勉強会では、 C++/CLI を native / managed の仲介役として紹介した。
- だとしたら、 C++ 、 C++/CLI 、 C# はそれぞれにいいところや得意とするところがあり、多言語開発環境の良さも、技術者の皆さんも分かっているんじゃないか。
- 皆さんも我々と一緒に欠点の指摘ではない C++ 、 C++/CLI 、 C# の適材適所を話し合っていきましょう。

C++、C++/CLI、C# の比較

- 生産性
- 拡張性
- 規模感
- 得意分野

C/C++ :生産性

- 既存資産をそのまま利用可能
- 個人のスキルに強く依存
- 言語仕様の複雑化
  - 多重継承
  - 例外
  - Template

C# :生産性

- RAD 環境で簡単に実装
- MVC パターン・基盤整備などに分離できる
  - 初心者から熟練者まで活躍できる
  - コンポーネントを購入してUIを拡充しやすい
- 言語仕様の複雑化
  - 例外
  - Generics
  - あまり使わない機能もある（ yield return とか）
- データベースとの親和性
  - DataSet
  - Linq

C++/CLI :生産性

- 既存資産をそのまま利用可能
- 個人のスキルに強く依存
- 言語仕様の複雑化
  - 多重継承
  - 例外
  - Template

ここまではC/C++と同じ。これに加えて

- 言語仕様の**さらなる複雑化**
  - Managed object
  - Generics

などなど、.NET とのハイブリッド構造　(Managed C++ よりマシ)

C/C++:拡張性

- OS に依存しないポータビリティ
- C は C で、C++ は C++ で作成可能
- 実行時イメージ(メモリイメージ)が同じであればクラスメソッドであっても追加できる拡張性の高さ
  - 下位互換を維持したまま拡張ができる
  - メンテナンス性が著しく下がる
    - 具体例:MFC4.2～6.0までの各バージョン

C# :拡張性

- 購入したりWebサイトから情報を得てコンポーネントを拡張できる
- 新技術のターゲット言語になりやすい
  - LINQ
  - AJAX
  - WPF・WCF・WF
  - SQL CLR
  - VSTO
- .NET Framework が Win32API ベースでなくなると C++ の代わりに基盤言語として存在可能か？

C++/CLI :拡張性

- 必要に応じて C , C++ , C++/CLI を自由に混在可
- native / managed 双方のライブラリが呼べる
- 拡張性は相当のキャパあるけど…やりすぎは禁物?

無理して使うもんじゃないよ

C/C++:規模感

- 少人数でのスペシャリストによる開発
- コアコンポーネントの開発
- 全体性能(レスポンスとパフォーマンス)を必要とするもの

C# :規模感

- 多人数での画面・帳票を多く含む開発
- UIベースで高速性能を求められないもの
- 画面・帳票とデータベースを中心とする業務
- Webによる多画面の開発
- 3層構造やSOAP・SOAなど多くの資源を必要とする開発

C++/CLI :規模感

- (C/C++と同じく)少人数でのスペシャリストによる開発
- nativeなコアコンポーネントと.NETとの**仲介役**
- IDEの生成するUIコードには問題あり
  → 「すべてをC++/CLIで実装」は**お勧めしない!**
  (WPFとほぼ－だし…)

C/C++ :得意分野

- OS やドライバとの直接的な連携部分
- OLE クライアント/サーバー/オートメーションサーバー
- Shell - Extension
- グローバルフック
- デバイスドライバ
- インストーラ
- Bootstrapper (インストーラ起動アプリ)

C# :得意分野

- なんといっても RAD 環境による簡単な開発
- 多人数でのチーム開発
- UIとデータベース中心の開発
- 新技術のおためしの場として

C++/CLI:得意分野

- なんといっても native / managed の**仲介役**
  - 既存 legacy ライブラリの .NET での利用
  - .NET 化のための薄いラッパー
- C/C++ コードの .NET への移植

  ↑ 文法は C# チックなC++
  - STL/CLR , marshaling library など、手駒は十分♪
- C++感覚で .NET Programming

まずは名脇役として

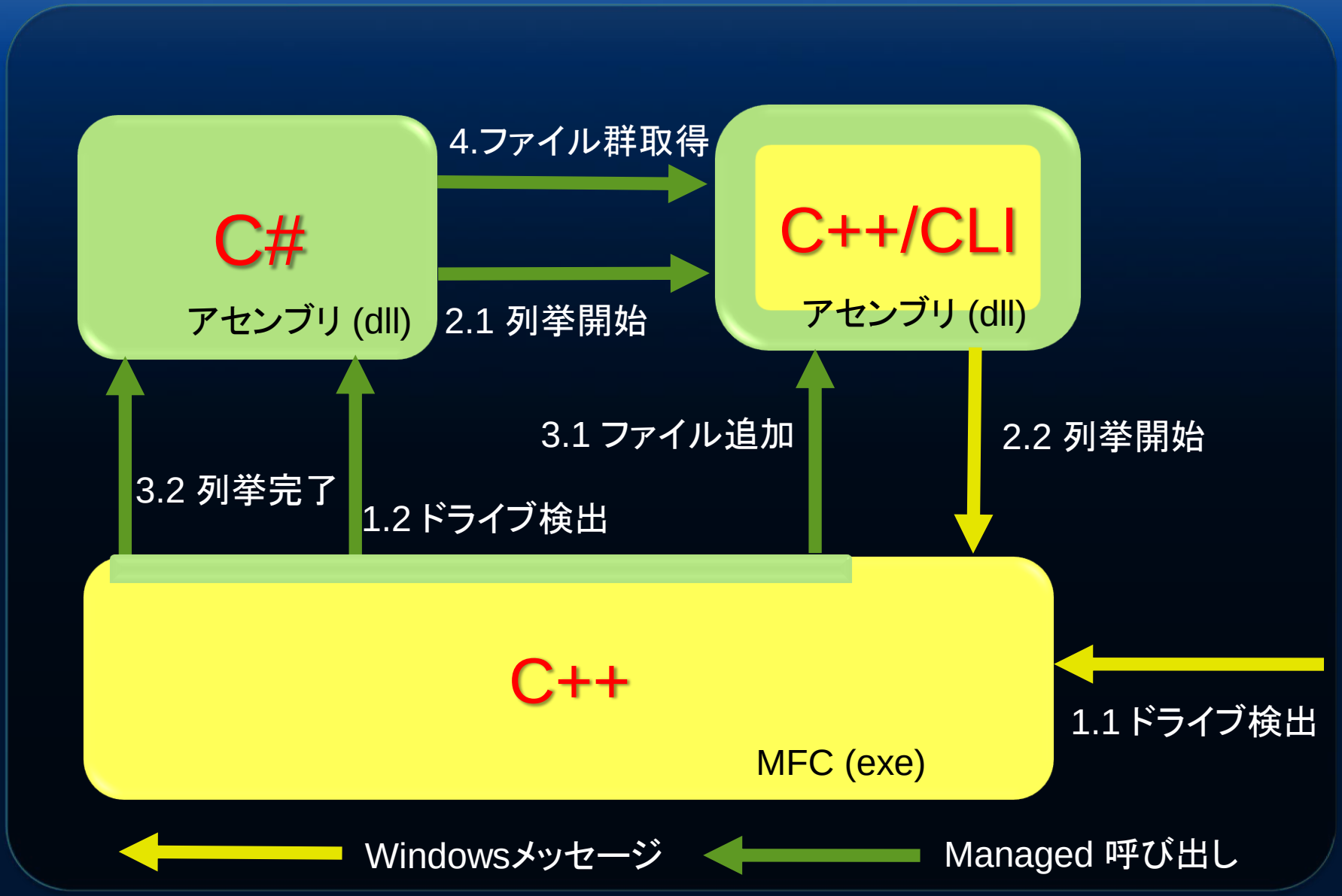
Microsoft Community Ring

DEMO

3人がこう思う使いどころ

とっちゃん	高萩 俊行	**Microsoft MVP Windows - SDK**
επιστημη	福田 文紀	**Microsoft MVP Visual C++**
えムナウ	児玉 宏之	**Microsoft MVP Visual C#**

DEMOのからくり


C#
アセンブリ (dll)
4.ファイル群取得
2.1 列挙開始
C++/CLI
アセンブリ (dll)
3.1 ファイル追加
2.2 列挙開始
3.2 列挙完了
1.2 ドライブ検出
C++
MFC (exe)
1.1 ドライブ検出
Windowsメッセージ
Managed 呼び出し

C++: サンプル抜粋1

```
class CMainFrame : public CFrameWnd
{
   // 省略…
   afx_msg BOOL OnDeviceChange(UINT
      nEventType, DWORD_PTR dwData);
};

BEGIN_MESSAGE_MAP(CMainFrame, CFrameWnd)
   // 省略…
   ON_WM_DEVICECHANGE()
END_MESSAGE_MAP()
```

C++:サンプル抜粋2

```
BOOL CMainFrame::OnDeviceChange(UINT
   nEventType, DWORD_PTR dwData)
{
   BOOL bResult =
      CFrameWnd::OnDeviceChange( nEventType,
            dwData );
   // メディアが挿入されたことをビューに通知
   return bResult;
}
```

C++:サンプル抜粋3

```
class CDevChangeHint : public CObject
{
public:
        CDevChangeHint( UINT nEventType,
                        DWORD dwUnitMask,
                              WORD nFlags );
        ~CDevChangeHint();
        bool          IsInserted() const;
        CString       TargetVolume() const;
        bool          IsVolumeTypeMedia() const;
        bool          IsVolumeTypeRemote() const;
        bool          IsVolumeTypePhysical() const;
};
```

C++:サンプル抜粋4

```
CDevChangeHint hint(nEventType,
                    pDBCV->dbcv_unitmask,
                    pDBCV->dbcv_flags );
CDocument* pDoc = GetActiveDocument();
pDoc->UpdateAllViews( NULL,
  WM_DEVICECHANGE, &hint );
```

C# : WinformからWPFをホスト

```
private ElementHost ctrlHost;
private MnowTechedBofWpf.TechedBofWpf
   techedBofWpf;

private void UserControl1_Load
(object sender, EventArgs e)
{
    ctrlHost = new ElementHost();
    ctrlHost.Dock = DockStyle.Fill;
    panelWpf.Controls.Add(ctrlHost);
    techedBofWpf = new
      MnowTechedBofWpf.TechedBofWpf();
    techedBofWpf.InitializeComponent();
     ctrlHost.Child = techedBofWpf;
}
```

C#：WPFでコードからImage操作

```
public void DoEmbossBitmapEffect()
{
  this.image1.BitmapEffect = new
     EmbossBitmapEffect();
}
public void SetBitmapFileName
   (string source)
{
  this.image1.Source = source;
}
```

C++/CLI: Native を Managed でラップ

```
class Native { … };

ref class Managed {
  Native* native;
public:
  Managed() { native = new Native(); }
 ~Managed() { this->!Managed(); }
 !Managed() { delete native; }
  …
};
```

- クラスの場合、ポインタに限る
- new/delete をお忘れなく
- auto_ptr<Native> 使用不可
- インスタンスのコピー時に注意!

C++/CLI: Managed を Native でラップ

```
ref class Managed { … };

class Native {
  gcroot<Managed^> managed;
public:
  Native() { managed = gcnew Managed(); }
 ~Native() { /* delete managed; は不要 */}
  …
};
```

- クラスの場合、直接メンバになれない
- **gcnew** をお忘れなく
- インスタンスのコピー時に注意!

#include <vcclr.h> しませう！

C++/CLI: DEMOでは…

```
ref class Proxy {
   HWND hWnd;  // PostMessage 先
   UINT Msg;   // PostMessage コード
   IList<String^> files; // 列挙ファイル群
public:
   void AddFile(wchar_t* file)
      { files.Add(gcnew String(file)); }
   IEnumerable<String^> GetFiles()
      { return files; }
   void StartEnumeration(Char drive)
   { PostMessage(hWnd,Msg,0,drive); }
   …
};
```

Microsoft Community Ring

あなたの思う使いどころ

C++、C++/CLI、C# の比較

- 生産性
- 拡張性
- 規模感
- 得意分野

Microsoft Community Ring

まとめ

適材適所、なればこそのVisual Studio
by *επιστημη*

とっちゃん
επιστημη
えムナウ

高萩 俊行 **Microsoft MVP Windows - SDK**
福田 文紀 **Microsoft MVP Visual C++**
児玉 宏之 **Microsoft MVP Visual C#**

Microsoft®

Your potential. Our passion.™

Prepare to make your mark.

Microsoft Tech·Ed 2007