

CADmeister ADK (アプリケーション開発キット)

CADmeister ADK (Application Development Kit)

大和田 義人

要約 CAD/CAM システムを利用する上で操作感や設計効率をより向上させて導入効果を高めるためには、顧客はシステムをカスタマイズして使用する。

2013年度にリリース予定のCADmeister ADKは、CADmeisterのカスタムアプリケーション開発を統合的にサポートするカスタマイズ用のツールキットである。CADmeister ADKの大きな特徴は、統合開発環境としてVisual Studioを使用できる点、開発プログラミング言語がVisual C++である点、高機能のアプリケーションが実現できる点、各種機能によって開発を容易にできる点である。

今後はCADmeister ADKによって、顧客のカスタマイズを容易に実現させると共に、サードパーティがアプリケーションを開発して顧客に提供することで、顧客は様々な種類のアプリケーションを使用できるようになる。

Abstract When utilizing CAD/CAM (Computer Aided Design/Manufacturing) system, customers customize the system before using in order to enhance the implementation effect by improving an operational feeling and design efficiency.

CADmeister ADK that is scheduled for release fiscal 2013 is the tool kit for customization that performs integrated supports in development of CADmeister custom applications. The main features of CADmeister ADK are as follows: 1) Visual Studio can be used as integrated development environment, 2) development language is Visual C++, 3) highly-functional application could be developed, 4) development work become easier by using various functions.

For the future, CADmeister ADK is expected to realize customer-led customization easily and provide third-party developed application to them, which enables the customers to use various types of applications.

1. はじめに

CAD/CAM (Computer Aided Design/Manufacturing: コンピュータ支援設計/製造) システムを利用する上で操作感や設計効率をより向上させて導入効果を高めるためには、顧客はシステムをカスタマイズして使用する。カスタマイズは簡単なものから複雑なものまで多岐にわたる。例えば、メニューの構成変更やショートカット登録をして利便性を高めたり、定型的な繰り返し作業を自動化して設計効率を向上させたりするような簡単なカスタマイズから、顧客独自の設計ノウハウを組み込んで設計品質を高めたり、CADデータを活かした業務フローを円滑にするために他のシステムと連携させたりする複雑なカスタマイズまである。市販されている一般的なCAD/CAMシステムは、内容や適用範囲に差はあるが、ほとんどがカスタマイズ機能を持っており、カスタムアプリケーションを開発するためのツールキットを提供しているシステムも少なくない。

日本ユニシス・エクセリューションズ株式会社（以下「UEL」と略す）は、自社で開発、および販売している3次元CAD/CAMシステム「CADmeister」をカスタマイズするツールキットとして、CADmeister Application Development Kit（以下「CADmeister ADK」と略す）を開発し、2013年度に正式リリースする予定である。CADmeister ADKを利用することによって、独自のコマンドを開発し、そのコマンド群をカスタムアプリケーションとしてCADmeisterにアドオンして標準アプリケーションと同じように使用できる。

本稿では、2章でCADmeister ADKの開発の背景、3章でCADmeister ADKの特徴、4章でCADmeister ADKのメカニズムについて述べる。

2. CADmeister ADKの開発の背景

CADmeisterは従来からCADmeister Customize Kit（以下「CCK」と略す）と呼ぶカスタムアプリケーションを開発できるツールキットを顧客に提供してきた。顧客はCCKを利用して主に標準アプリケーションのコマンドを一度に複数実行するような簡単なコマンドを開発して設計効率を向上させていた。しかし、CCKには、以下のような課題があった。

- ・開発プログラミング言語がスクリプト言語と呼ぶCADmeister独自の専用言語である
- ・統合開発環境がない
- ・高機能のアプリケーションを実現できない

これらの課題を解決し、標準アプリケーションを補って顧客独自の設計ノウハウを活かせるカスタムアプリケーションを簡単に開発できる新たなツールキットの提供が望まれていた。また、サードパーティが開発したカスタムアプリケーションが顧客に提供されることによって、顧客が様々な種類のアプリケーションを使用できるようになることも望まれていた。以上がCADmeister ADKを開発することになった背景である。

3. CADmeister ADKの特徴

本章では、前章で述べたCCKの課題を解決し、カスタムアプリケーション開発を強力に支援するCADmeister ADKの特徴的な機能について述べる。

3.1 CADmeister ADKのカスタマイズ適用範囲

CADmeister ADKによるカスタマイズの適用範囲は、顧客やサードパーティが独自のコマンドを開発することと、開発したコマンド群をカスタムアプリケーションとしてCADmeisterにアドオンすることである。

カスタムアプリケーションの開発、および使用の形態として以下の三つを想定している。

- ・顧客が開発し、顧客自身で使用する
- ・UELの受託開発パートナーが開発し、委託元の顧客が使用する
- ・UELの商品づくりパートナーが開発し、一般の顧客が使用する

受託開発パートナーとは、CADmeisterの顧客からカスタムアプリケーション開発を受託できる権利をUELが与えたサードパーティのことである。また、商品づくりパートナーとは、カスタムアプリケーションを開発し、CADmeisterの顧客へ販売できる権利をUELが与えたサードパーティのことである。顧客はカスタムアプリケーションを使用したい場合、顧客自身で開発するか、受託開発パートナーに発注して開発するか、商品づくりパートナーから購入す

るかを選択できる。いずれの形態でも CADmeister ADK で利用できる機能に差はなく、次節以降で述べる機能を全て利用できる。

3.2 統合開発環境

CADmeister ADK の各種機能は Microsoft 社の統合開発環境である Visual Studio のプラグインとして提供される。その結果、カスタムアプリケーション開発の統合開発環境として Visual Studio を使用でき、プログラムコードのコンパイルやリンクに加え、CADmeister へのメニュー登録やコマンド登録等の開発作業を一貫して簡単に行うことができる。Visual Studio は Windows アプリケーション開発ではデファクトスタンダードの統合開発環境であるため、専用の開発方法を習得する必要はない。

カスタムアプリケーション開発専用のテンプレートやウィザードも充実しており、開発の利便性を高める。カスタムアプリケーション用の Visual Studio のプロジェクトを作成する場合に専用のテンプレートを選択すれば、インポートライブラリやインクルードディレクトリ、生成ファイルの出力先等のほぼ全ての初期設定が自動的に施され、プロジェクト作成時の煩雑な設定が不要になる。また、カスタムアプリケーションにコマンドを追加する場合も専用のテンプレートを選択すれば、ウィザード形式でコマンドの設定を簡単に定義できる。コマンドの設定項目はコマンド名やコマンド動作モード、入出力パラメータ、パラメトリックオプション等である。更に定義したコマンドの設定をもとに初期のプログラムコードが自動的に生成される。コマンドの設定を各設定ファイルで個別に定義する必要はなく、プログラムコードを何もない状態から作成する必要もないので、CADmeister ADK ではコマンド開発の初期作業が大幅に軽減される。

3.3 開発プログラミング言語

CADmeister ADK の開発プログラミング言語は Microsoft 社の Visual C++ である。Visual C++ は Windows アプリケーションの一般的な開発プログラミング言語として広く普及し、UEL での CADmeister のアプリケーション開発でも使用しているため採用した。Visual C++ での開発経験があれば、カスタムアプリケーション開発をスムーズに開始できる。Visual C++ に付属しているライブラリである MFC (Microsoft Foundation Classes) を利用することによってファイル入出力操作や文字列操作、GUI 操作等を簡単にプログラミングできる。

Visual C++ で記述されたプログラムコードのデバッグには Visual Studio のデバッグ機能を全て使用できる。ブレークポイントの設定や変数のウォッチ、コールスタックの参照等を行うことができるためデバッグ作業の効率は良い。Visual Studio のデバッグ機能に加えて CADmeister のログ出力機能も使用できる。ログ出力機能を利用することによって、Visual Studio がインストールされていない端末でもデバッグに必要な情報を収集できる。

3.4 アプリケーションの機能実現性

図 1 に示すとおり、CADmeister ADK では標準アプリケーションのコマンドやプリミティブ機能だけでなく、更にその下位の機能も呼び出すことができる。プリミティブ機能とはスクリプト言語で記述されたコマンドのプログラムから呼び出されることを目的として作成された C 言語の API (以下「C-API」と略す) である。標準アプリケーションのコマンドはいくつか

のプリミティブ機能呼び出すことで実現されている。

標準アプリケーションのコマンドやプリミティブ機能呼び出すだけでも、複数のコマンドを一括実行させて設計効率を向上させたり、設計要件が満たされているか確認して設計品質を向上させたりできる。既存のプリミティブ機能では機能要件を満たせない場合は独自のプリミティブ機能を作成して呼び出すこともできるため、高機能のカスタムアプリケーションを実現できる。例えば、図形/形状処理を利用して独自の形状変形エンジンを構築し、利用できる。また、他ベンダのシステム API を呼び出せるため、PDM*¹ や PLM*²、CAE*³ 等のシステムと CADmeister を自由に連携させて相乗効果を生み出すことができる。

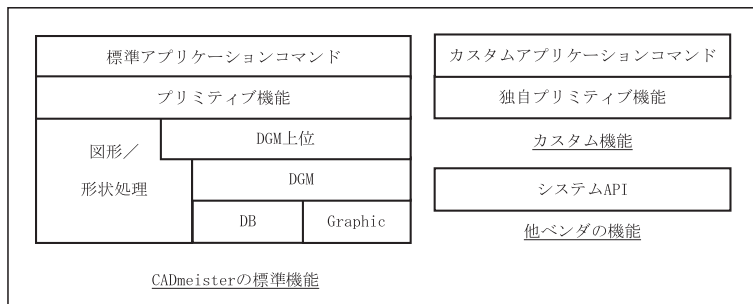


図1 CADmeister ADK で呼び出し可能な機能の範囲

3.5 オブジェクト指向 API

従来はプリミティブ機能をC-APIとして提供してきたが、開発プログラミング言語がオブジェクト指向の Visual C++ になり、C-APIのままでは利便性が悪いため、プリミティブ機能はオブジェクト指向 API としても提供する。幾何要素やレイヤ、グループやスケッチ等 CADmeister 上の概念データはそれぞれクラスとして定義し、プリミティブ機能はそのメソッドになる。

図2にグループのクラス図の例を示す。グループの属性である名前や注記、色番号等はグループクラスのプロパティとなり、グループメンバの追加や削除の操作はメソッドとなる。クラスになることによって、該当するプリミティブ機能の検索がC-APIに比べて容易であるため開発効率が向上する。

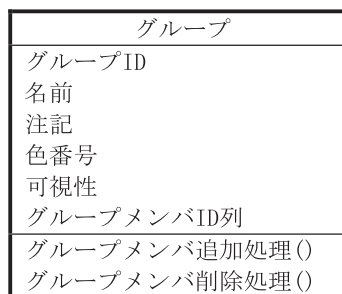


図2 グループのクラス図

3.6 自動 GUI 作成機能

CADmeister ADK には、コマンドの操作パネルやポート上のハンドル等の GUI をプログラ

ミングしなくても動的に自動で作成できる機能がある。ハンドルとは Office アプリケーションのアンカーポイントに似た機能を持つ GUI で、ポート上でマウスを利用した直感的な操作によって、位置や長さ、角度等のパラメータの入力を可能にする。

一般的な Windows アプリケーションのダイアログのように、コマンド毎に操作パネルのリソースを定義して保持させることもできるが、この場合は操作パネルのコントロールの大きさや位置を綿密に定義しなければならない。自動 GUI 作成機能を使用すれば、図 3 に示す XML 形式のファイルに GUI と入力パラメータの関係を定義しておくだけで、コマンド実行時に図 4 に示す操作パネルとハンドルを動的に自動で作成させることができる。

作成後の操作パネルやハンドルに関する処理も全て CADmeister ADK が受け持つため、開発者は GUI に関する処理をプログラミングする必要は一切ない。球を作成するコマンドを例にし、図 3 と図 4 を用いて詳細を述べる。球の作成方法として、中心点と半径から作成する方法と円もしくは円弧を参照して作成する方法の二通り定義してある。本来は次のような GUI に関する処理をプログラミングする必要がある。

- ・球の作成方法の選択によって操作パネルやハンドルを差し替える
- ・球の半径としてゼロ以下の値が入力できないようにする
- ・ポートで中心点と半径のハンドルが操作された時に相対する操作パネルの値を変更する
- ・操作パネルで中心点や半径の値が変更された時に相対するハンドルを更新する

CADmeister ADK の自動 GUI 作成機能を使用すれば、設定された入力パラメータに従って球を作成するコードをプログラミングするだけでコマンドを開発できる。コマンド開発において、意外に開発工数がかかって不具合が混入し易い GUI に関する処理のプログラミングを省くことができるため、開発工数を削減でき、品質も向上させることができる。

```
<Dialog caption_id="0">
  <Group name="CREATE_METHOD" status="open" caption_id="1">
    <Select name="SELECT_CREATE" relation="i_niCreateMethod" caption_id="11" guidance_id="110">
      <Items>
        <Item name="ITEM_CENTER_RADIUS" value="0" caption_id="111" tooltip_id="1110"/>
        <Item name="ITEM_CIRCLE_ARC" value="1" caption_id="112" tooltip_id="1120"/>
      </Items>
    </Select>
  </Group>
  <Group name="CREATE_INPUT" status="open" caption_id="2">
    <Locate name="CENTER_POINT" relation="Center" caption_id="21" guidance_id="210">
      <Reference name="SELECT_CREATE">
        <Condition value="0"/>
      </Reference>
      <Handle type="free"/>
    </Locate>
    <Length name="RADIUS" relation="Radius" caption_id="22" guidance_id="220" minus="false">
      <Reference name="SELECT_CREATE">
        <Condition value="0"/>
      </Reference>
      <Handle type="freeEnd">
        <Locate relation="i_lcCenterPoint"/>
      </Handle>
    </Length>
    <Hit name="CIRCLE" relation="i_idCircle" caption_id="23" guidance_id="230">
      <Reference name="SELECT_CREATE">
        <Condition value="1"/>
      </Reference>
    </Hit>
  </Group>
</Dialog>
```

図 3 球コマンドの GUI 定義

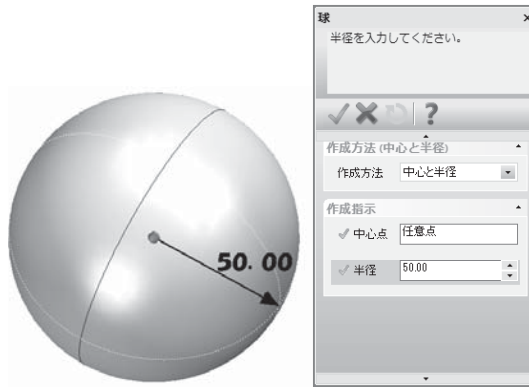


図4 図3の定義によって自動作成される GUI

4. CADmeister ADK のメカニズム

本章では CADmeister ADK のメカニズムについて述べる。CADmeister ADK は内部的には様々なユーティリティや機能を利用している部分が多いが、それらをカプセル化して使い易いように工夫している。また、Visual Studio をカスタマイズして容易にカスタムアプリケーションを開発できるようにしている。

4.1 開発時のメカニズム

統合開発環境だけでカスタムアプリケーションの開発が完結できるようにするため、プログラムコードのファイルだけでなく、アプリケーションに関するファイルも全て統合開発環境で管理している。カスタムアプリケーションの Visual Studio のプロジェクトは図5に示すとおりコマンド毎のプログラムコード、アプリケーションメニューファイル、アプリケーション登録ファイルから構成される。

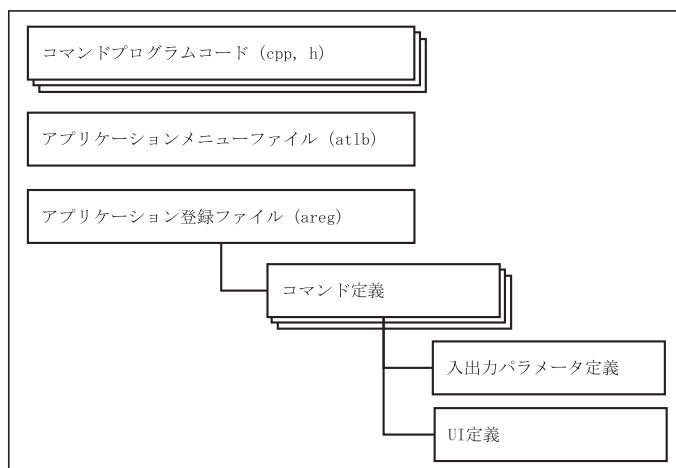


図5 アプリケーションプロジェクトの構成

4.1.1 コマンドプログラムコード

コマンドプログラムコードは Visual C++ でプログラミングされた各コマンドクラスのプロ

グラムコードである。プロジェクトのビルド時に Visual Studio の標準機能によってコンパイル、リンクされ、カスタムアプリケーション DLL として所定のフォルダに作成される。一つのカスタムアプリケーションに対して一つの DLL が作成される。

4.1.2 アプリケーションメニューファイル

アプリケーションメニューファイル (拡張子 atlb) は CADmeister にカスタムアプリケーション用のツールバーを登録するためのファイルである。プロジェクトのビルド後に自動的に CADmeister ADK のユーティリティが動作するように Visual Studio をカスタマイズしており、このユーティリティによってツールバーが登録される。図 6 にアプリケーションメニューファイルと登録されるツールバーの例を示す。カスタマイズには「構成プロパティ/ビルド イベント/ビルド後のイベント」機能を利用しているが、ビルドが正常に終了した場合にコマンドラインでコマンドを実行できる機能である。

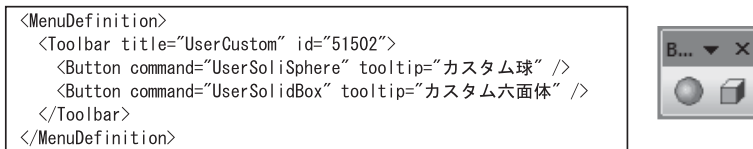


図 6 アプリケーションメニューファイルとツールバーの例

4.1.3 アプリケーション登録ファイル

アプリケーション登録ファイル (拡張子 areg) はカスタムアプリケーションに含まれる各コマンドを定義するファイルである。コマンド定義は入出力パラメータ定義と GUI 定義からなる。入出力パラメータ定義の例を図 7 に示す。作成方法 1 では入力された中心点と半径から球を作成し、作成方法 2 では入力された円・円弧から球を作成する。出力パラメータは作成された球の ID である。GUI 定義の内容は図 3 に示したとおりである。



図 7 入出力パラメータ定義の例

アプリケーションビルド時にアプリケーション登録ファイルがCADmeister ADKのユーティリティによってコンパイルされるように Visual Studio をカスタマイズしている。カスタマイズには「構成プロパティ/カスタム ビルド ステップ」機能を利用しているが、これは特定のファイルに対してカスタム ビルドを指定できる機能である。

図8に示すとおり、アプリケーション登録ファイルからユーティリティによってコマンド登録ファイルとコマンドディスパッチプログラム、各コマンドのスクリプト言語プログラムコードが生成される。

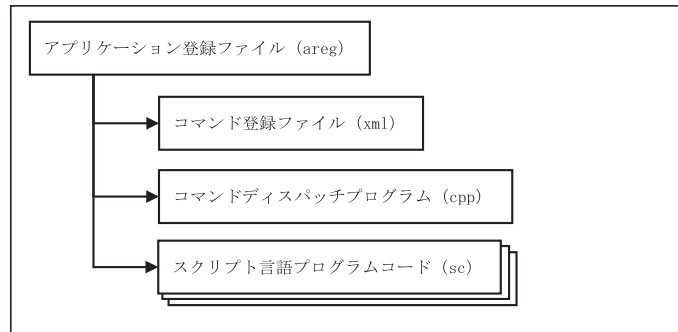


図8 アプリケーション登録ファイルから生成されるファイル

コマンド登録ファイルはCADmeisterのコマンドとして登録するために必要なファイルで、所定のフォルダにカスタムアプリケーション一つに対して一つ生成される。

コマンドディスパッチプログラムはDLLをカスタムアプリケーションDLLとして動作させるために必須のプログラムであり、カスタムアプリケーションDLLに組み込まれる。ツールバーからコマンドが実行された時にDLL内の該当するコマンドクラスをインスタンス化したり、コマンドが終了した時にインスタンスを破棄したりするプログラムである。

スクリプト言語プログラムコードはコマンドとして必要なスクリプトモジュールを生成するために必要となる。ユーティリティによってコマンド定義からスクリプト言語プログラムコードを生成し、コンパイル、およびリンクまでをバックグラウンドで実行してスクリプトモジュールを生成する。

4.2 コマンドの実行と履歴再生のメカニズム

CADmeister ADKは前節で述べたとおりスクリプトモジュールを生成している。CADmeisterのコマンドフレームワークでは、入力処理とメイン処理にはコマンドクラスを利用しているが、パラメトリック機能の履歴作成処理や入力復元処理にはスクリプトモジュールを利用しているためである。

コマンドの実行時と履歴再生時の処理フローを図9に示す。コマンド実行時の処理フローは以下のとおりである。

- (1) アプリケーションDLLがロードされ、コマンドクラスでパラメータを入力させる（入力処理）
- (2) パラメータの入力が確定するとコマンドクラスからスクリプトモジュールを呼び出す

- (3) スクリプトモジュールで入力パラメータをもとにして履歴データを作成する (履歴作成処理)

※ノンパラメトリックコマンドの場合は履歴を作成しない

- (4) スクリプトからコマンドクラスを呼び出す
 (5) コマンドクラスで入力パラメータに従って実際に形状を作成する, 変形する等, コマンドの根幹となる処理を実行する (メイン処理)

履歴再生時の処理フローは以下のとおりである.

- (6) 履歴データからスクリプトモジュールの入力パラメータを復元する (入力復元処理)
 (4) スクリプトからコマンドクラスを呼び出す
 (5) コマンドクラスでメイン処理を実行する

CADmeister ADK を使用してコマンドを開発する場合にプログラミングする必要があるのは入力処理とメイン処理のみである. その他はCADmeister ADK によって自動的に処理しているため不要である. 自動 GUI 作成機能を利用する場合は入力処理も不要となる. 以上のようにして, CADmeister ADK では開発者がプログラミングする処理を減らし, カスタムアプリケーション開発の簡略化を実現している.

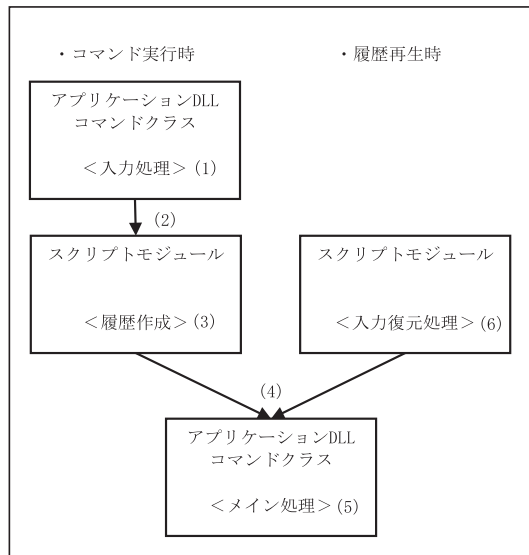


図9 コマンド実行時と履歴再生時の処理フロー

5. おわりに

CADmeister ADK の開発において一番の課題は, 顧客やサードパーティの開発者が一般的な開発プログラミング言語で簡単にCADmeister のカスタムアプリケーションを開発できるツールキットにすることであった. 開発プログラミング言語には Visual C++ を採用し, Visual Studio を統合開発環境として利用することによって解決できたと評価している.

今後のCADmeister ADK の開発計画として XML 形式ファイルの専用 GUI エディタと開発

プログラミング言語の強化を予定している。前者によって XML 形式ファイルであるアプリケーションメニューファイルやアプリケーション登録ファイルを簡便に作成し修正できるようになる。また、後者によって Visual C++ に加えて Visual Basic や Visual C# 等の Microsoft .NET Framework 系の開発プログラミング言語でもカスタムアプリケーションを開発できるようになる。

UEL が CADmeister のアプリケーションを開発する際にも CADmeister ADK を利用していく予定である。自社でも使い込んでいくと共に、顧客やサードパーティからの意見を反映させながら CADmeister ADK の性能を改良し、CADmeister のカスタマイズ性をより一層向上させることによって、顧客やサードパーティのビジネスの成功に貢献していく所存である。

-
- * 1 Product Data Management : 製品データ管理.
 - * 2 Product Lifecycle Management : 製品ライフサイクル管理.
 - * 3 Computer Aided Engineering : コンピュータ支援解析.

執筆者紹介 大和田 義人 (Yoshito Owada)

1999 年に日本ユニシス株式会社入社。2006 年より日本ユニシス・エクセリョーションズ株式会社に出向中。入社後から現在に至るまで一貫して 3 次元 CAD/CAM システムのフレームワーク開発に従事。

