

UML2.0対応 組込み系モデリングツール TAU/Developer のご紹介

日本テレロジック(株)

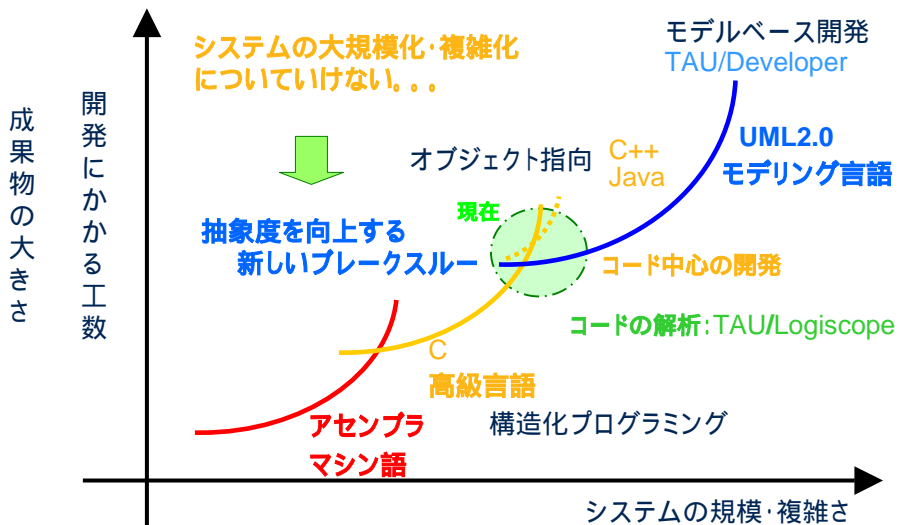
URL www.telelogic.com/jp

メール Info@telelogic.co.jp

Tel 03-5427-8900 (代表)

Telelogic

開発の限界



UML 2.0 とは？

- UML 2.0 は、システムの仕様作成、設計、文書化に適した UML (Unified Modeling Language) の重要なバージョンアップである。
- UML 2.0 = UML1.4 + SDL2000 + MSC2000 + その他(OCLなど)
- 実行可能なモデルの構築と動作の詳細な記述に対するサポート
- ツール間でのダイアグラムのデータ交換が可能(XMI)

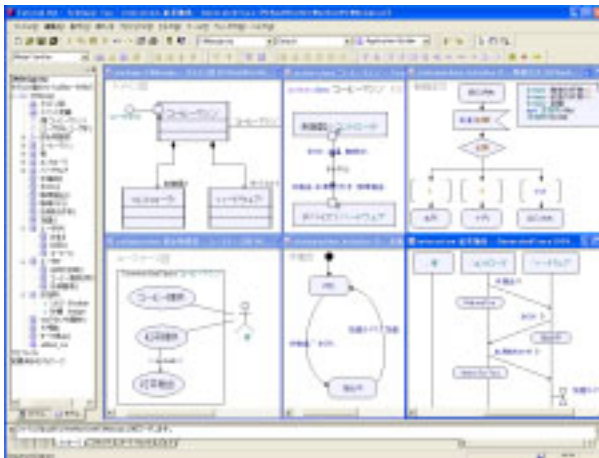
UML言語そのものは以下の4つのパートからなる

- Infrastructure : プロファイルを作れる(標準化したものを使えばよい)
- Superstructure : UMLの記述方法(SDLなどを追加)
- Diagram Interchange : XMI2.0(ツールが対応すれば良い)
- OCL : OCL2.0(滅多に使われない)



UML2.0:SDLの採用でモデルベース開発が現実的に！

TAU/Architect および Developer



- テレロジック社 が提供する UML2.0 準拠統合開発環境
- 特定の метод論や開発プロセスには非依存
- C++/Cフルコード生成
- Javaスケルトン生成
- シミュレーション機能
- COM/Tclによるカスタマイズ
- オンラインチェックと自動修正
- 任意の位置にコメント記入可
- 構成管理ツールとの連携(SCCI)
- 要件管理ツールとの連携

サポートしているダイアグラム

TAU2.3ではUML2.0の13種類のダイアグラムのうち、以下の10種類をサポートしている

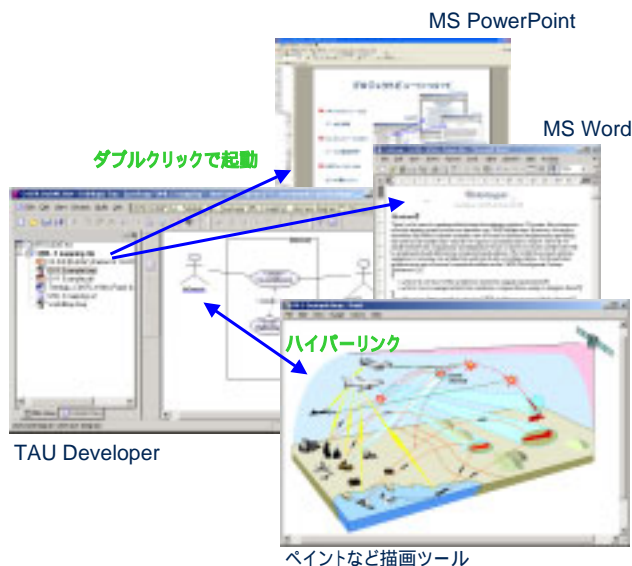
- ユースケース図
- シーケンス図
- アクティビティ図
- 相互作用概観図
- クラス図
- コンポジットストラクチャ図
- パッケージ図
- コンポーネント図
- 配置図
- 状態マシン図

将来対応予定のダイアグラム

- コミュニケーション図
(旧: コラボレーション図)
- オブジェクト図
- タイミング図

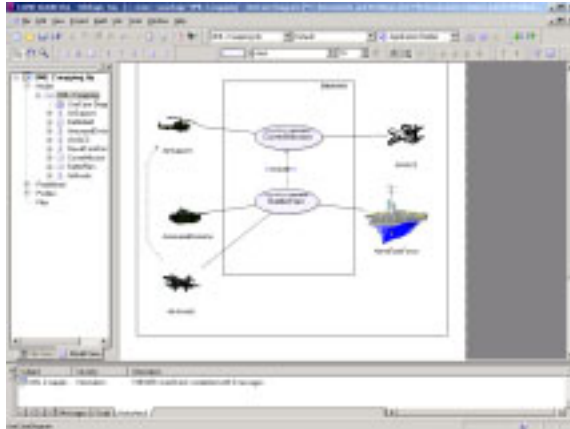
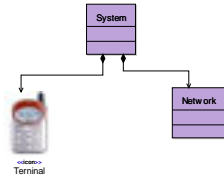
UML以外の文書の取り込み

- TAU Developer のプロジェクトビューから全ての関連データにアクセス可能！
- 外部ファイルとUMLモデル間をリンク



ユーザー定義のアイコン

- ビットマップの利用など、アイコンをユーザー定義でもできるようになり、モデルの視認性を向上します。



TAU独自の強力な編集機能

オート・コンプリート (自動補完)

1. 文字を入力します

Li|

2. Ctrl+スペースを押す

Li|ft

3. Ctrlキーを押したままスペースキーを押すと次の候補が表示される

LiftController

オンラインチェック

構文 (Syntax) エラー

```
Integer myInt ▲? 12;
myInt = 11;
```

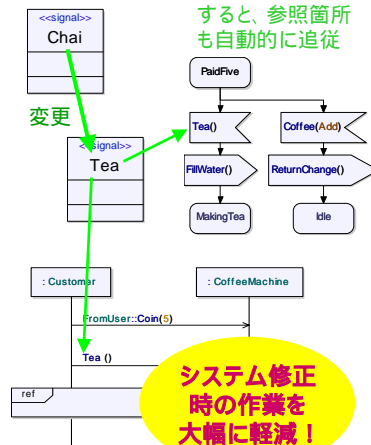
意味 (Semantic) エラー

```
Integer myInt = 12;
myint = 11;
```

常時1000以上のルールでチェック!

自動修正

定義箇所を変更すると、参照箇所も自動的に追従



システム修正時の作業を大幅に軽減!

モデルの記述段階でエラーを防止!

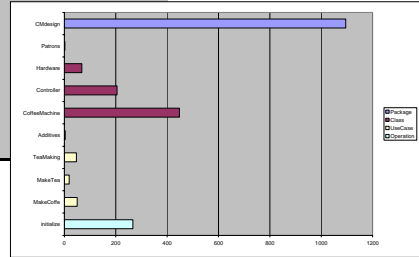
カスタマイズ用API

■メタモデルや各種機能へアクセスするAPIを提供

- C++, Java, Visual Basic, C# などから利用できるCOM API
- 詳細にプログラミングできるTcl API

■バッチ処理にもインタラクティブにも利用可

Component Name	CMdesign Kind	MakeCoffe Package Root	CoffeeMachine UseCase UserDrinks	Patrons Class CMdesign	initialize Operation Controller	initialize Operation Hardware
Operations	6	1	0	0	1	2
Attributes	-	-	2	0	-	-
Classes	10	0	1	1	0	0
Elements	1095	50	448	2	202	65
Definitions	65	1	5	1	14	5
Diagrams	11	1	1	-	-	-
Symbols per Diagram	13	6	5	-	-	-
Average	1	0	0	1	0	0
Maximum depth of inheritance tree	6	0	0	0	0	0
Maximum cyclomatic complexity	16	0	0	0	0	0
Number of states	2	0	-	-	-	-
Average Cyclomatic complexity	0,1	-	0	0	0	0

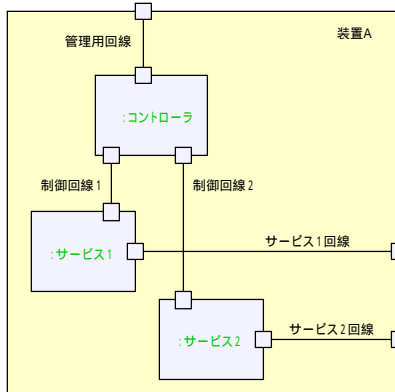


カスタマイズ例: モデルのメトリクス解析

- HTML およびCSV フォーマットへ出力
- ユーザー定義ルールの追加

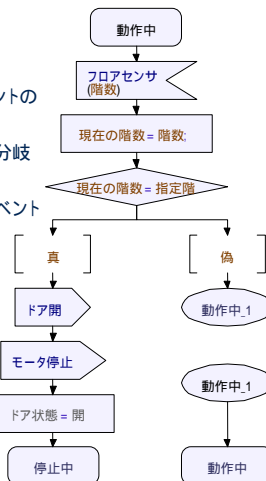
効果的なレビュー (1/2)

1. 可読性

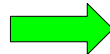


- 状態Aで取れるイベントの種類は?
- その遷移はいくつに分岐する?
- その遷移でどんなイベントを送信する?

- いくつかのサブシステムに分解されている?
- どのサブシステムとどのサブシステムがやりとりを行う?



- モジュール構造が一目で理解できる
- 詳細設計が理解できる



**レビューが成立する
目に見える設計成果物**

生産性の向上(2/2)

2. 自動化

コード生成

商業用途に実績のあるコードジェネレータを採用

- 高品質 & 高信頼性
- モデル=コード が可能
- ロイヤルティフリー
- ランタイムライブラリのソースコード提供

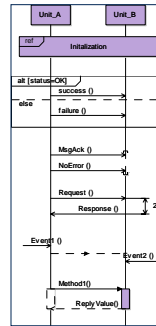


検証シナリオのスク립ト化



- リグレッションテストの自動化
- イベントの順番を一部入れ替えるのが容易

アクティブモデラー



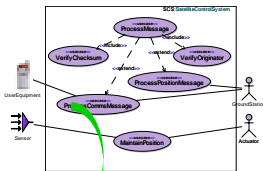
必要な定義(クラスなど)を自動生成

- シーケンス図から
- コンポジットストラクチャ図から
- etc.



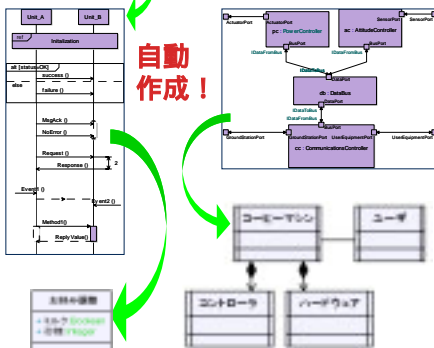
開発効率の向上!

アクティブ モデラー



- 高レベルのダイアグラムで作成開始

- ユースケース図、シーケンス図、アーキテクチャ(コンポジットストラクチャ)図



- 必要なモデル要素の作成を自動化

- 必要とされるクラス、動作、信号、ポートなどを自動定義

モデリング作業の大幅スピードアップ!

自動コード生成



- クラス図
- ステートマシン図
- コンポジット構造図

UMLモデルからコードを自動生成

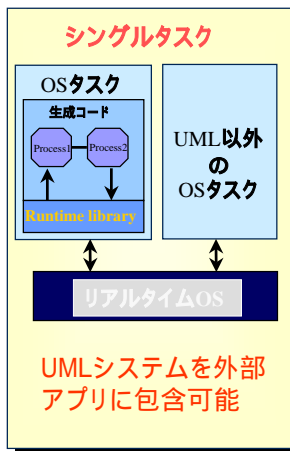
- 100% C++
- スケルトンJava
- 100% C



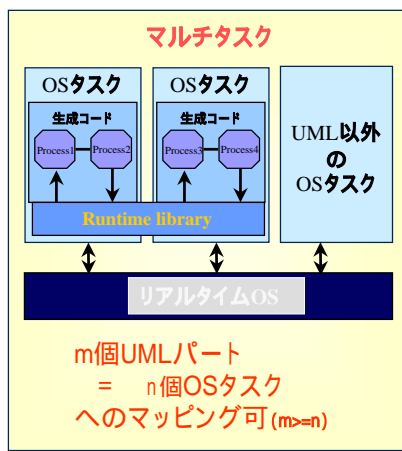
IDEとの統合

- .Net
- Eclipse
- Sun Java Studio、など

リアルタイムOSへのポーティング



1 UMLシステム
= 1 OSタスク



1 UMLシステム
= 複数のOSタスク

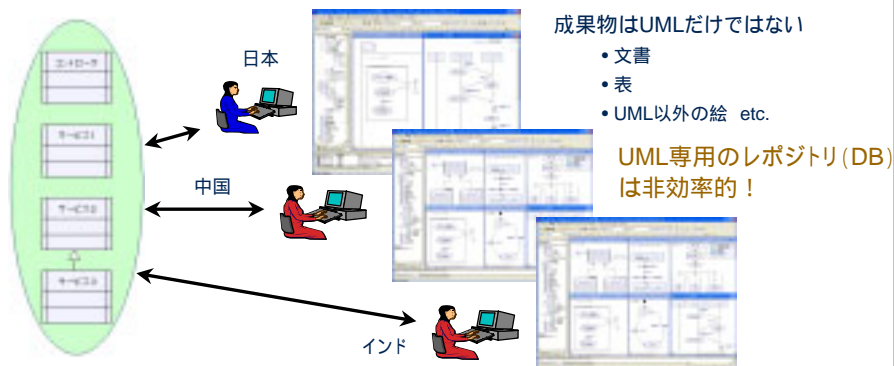
Runtime library
はソースコードを
提供

- Embedded Linux
- μiTRON4.0
- VxWorks
- ...etc

ターゲットデバッガ
とのツール連携

- microVIEW-PLUS
- ...etc

分散開発への対応



成果物はUMLだけではない

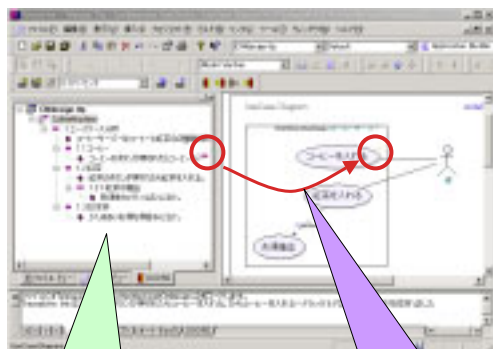
- 文書
- 表
- UML以外の絵 etc.

UML専用のレポジトリ(DB)は非効率的!

- 拠点間で成果物を共有する
 - バージョンの整合性の確保
- ➔ 分散開発に対応した構成管理ツールでUMLの成果物も一緒に管理する

XML形式ファイルで保存 どの構成管理ツールでも管理可能
SCCI対応により、TAU上からCheck-in/-outなどの操作が可能

トレーサビリティの確保(1/3) 1. 仕様書とUMLモデルのリンク



- CMMなど要件管理は必須になる
 - 要件がすべて(必要十分条件)
- 管理された要件を開発に利用する
 - 開発のトリガーは要件

**UMLだけでは
仕様書にならない!!**

要件管理ツール内の仕様書をUML開発環境で確認できる

ドラッグアンドドロップで要件からモデルへリンクを作成する

モデルと要件の関連要件とモデル要素にリンクをはり、モデルを要件管理下におく

トレーサビリティの確保(2/3)

2. UMLも含めた影響分析

仕様書の変更箇所

影響するモデル要素

DOORS上のUMLモデルIDから、対応するTAU上のモデル要素へジャンプ可能

さらに影響するモデル要素

変更時の影響箇所をモデルを含めて即座に把握可能！

トレーサビリティの確保(3/3)

3. モデルの変更管理

仕様書 DOORS

1つの変更要求であちこちが変わる... 複雑なシステムほど万全の管理が必要!

変更管理ツール SYNERGY/Change

いつ、どんな理由によって、どんな変更を加えるのか

構成管理ツール SYNERGY/CM

どんな変更要求に応じて、いつ、誰がどのファイルをどう変更したか

グラフィカルな比較&マージ

- 差分の容易な把握が可能

UML2.0開発環境 TAU/Developer

TAU Generation 2 とツール連携

