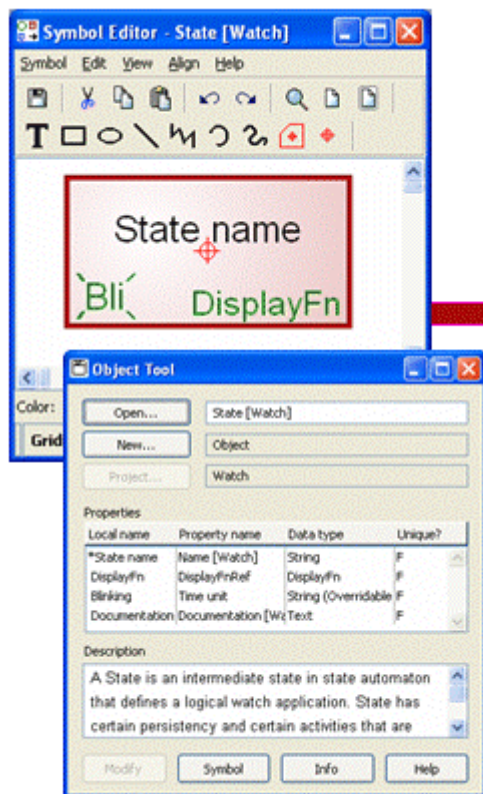


## < MetaEdit+ 概説書 > D S M環境構築、モデル開発、コード・仕様書の自動生成

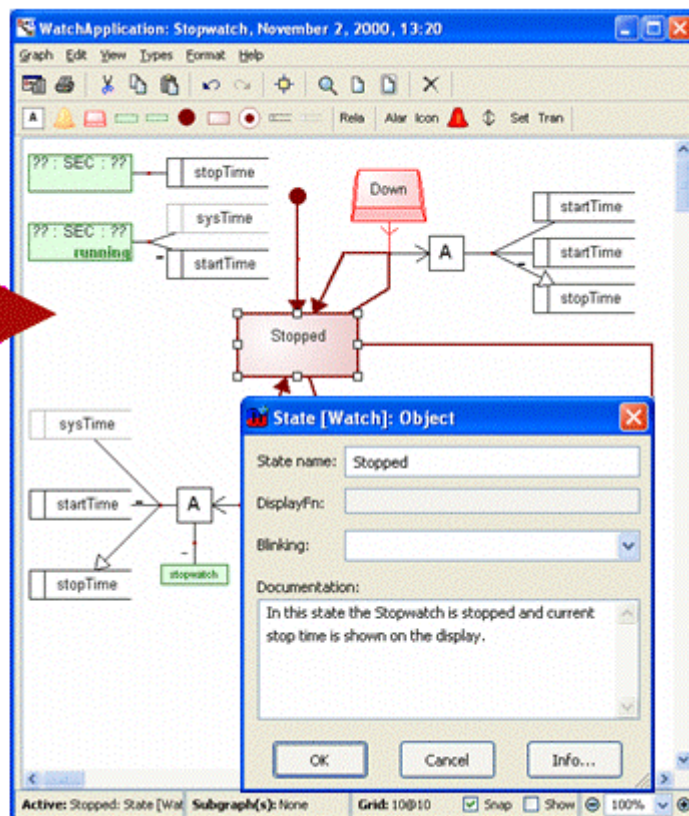
### MetaEdit+ DSM 環境

MetaEdit+は、完全なコードをモデルから直接生成させることで生産性の向上を求める開発者のためのツールです。エキスパートな開発者がモデリング言語を設計するためのツールが MetaEdit+ Workbench で、設計された言語を使って、他の開発者がモデルを作成する為のツールが MetaEdit+ です。

MetaEdit+ Workbench  
独自のモデリング言語を設計する



MetaEdit+  
モデリング言語の使用



#### MetaEdit+ Workbench

は、独自のモデリング言語を設計する為のツールです。言語は、言語のコンセプト、ルール、表示方法及びコード生成方法で定義され、その定義はメタモデルとして、MetaEdit+のレポジトリに保存されます。

Workbench のライセンスは、通常エキスパートな開発者向けの本又は数本で十分です。

#### MetaEdit+

は、Workbench で設計されたモデリング言語を使用して、モデルを作成するためのツールです。ダイアグラムエディター、ブラウザー、ジェネレータ、マルチユーザーサポート等の機能を提供します。

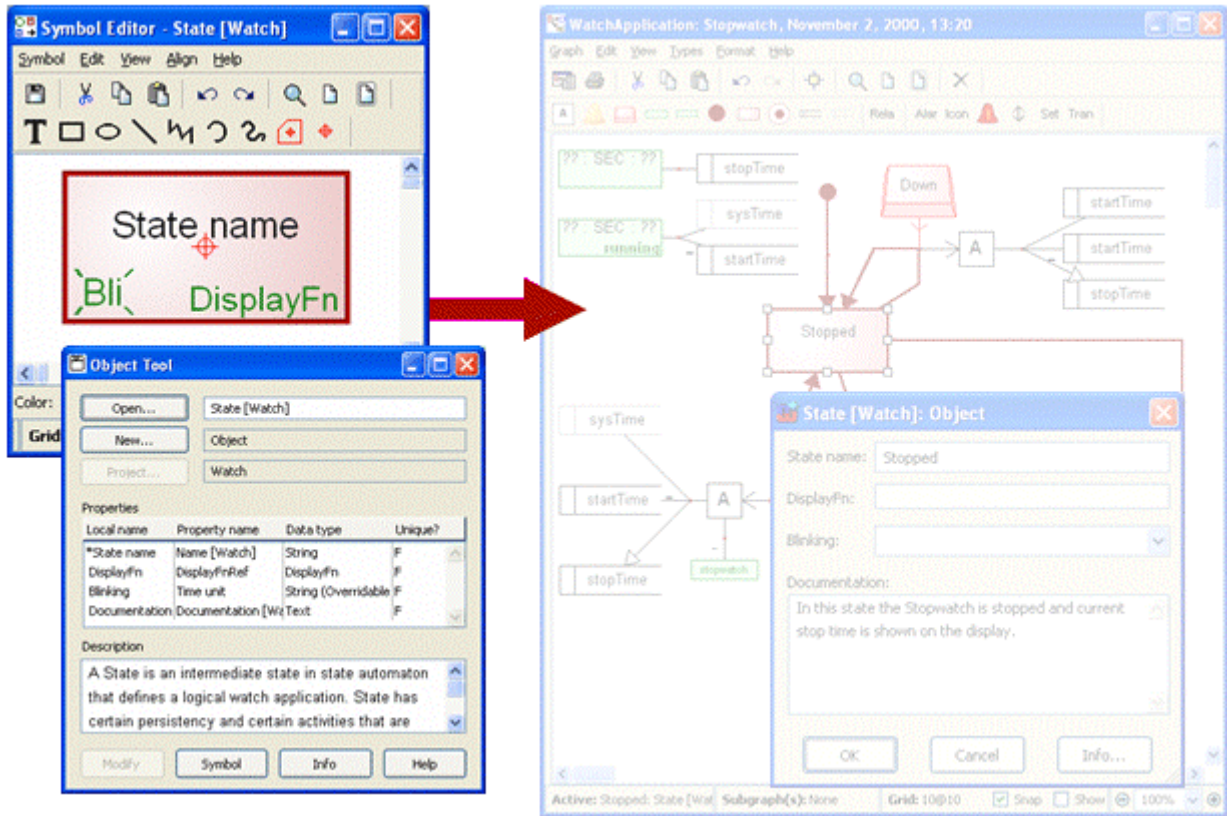
MetaEdit+ のライセンスは、このモデル作成ツールを使用して製品を開発する開発者毎に必要です。

## 《独自のモデリングツールを構築する》

MetaEdit+ Workbench を使えば、独自のモデリングツールを簡単に構築できます。モデリング言語を定義しさえすれば、直ぐに MetaEdit+ でモデルベースの開発が始められます。

独自のモデリング言語を設計する

モデリング言語を使用する



MetaEdit+ Workbench は、シンプルでありながら強力な、メタモデリング言語と独自のモデリング言語の設計ツールを提供します。これらのツールを使用することで、言語のコンセプト、それらのプロパティ、対応したルール、シンボル、言語のチェックレポート、ジェネレータの定義が簡単に行えます。再利用可能な非常に多くのライブラリがある為、導入後直ちに作業を始められます。MetaEdit+によるオブジェクト指向のメタモデル作成手順で、迅速で直感的、しかも費用効果の高い開発が行えるようになります。

## ステップ1：ドメイン固有のコンセプトを定義する

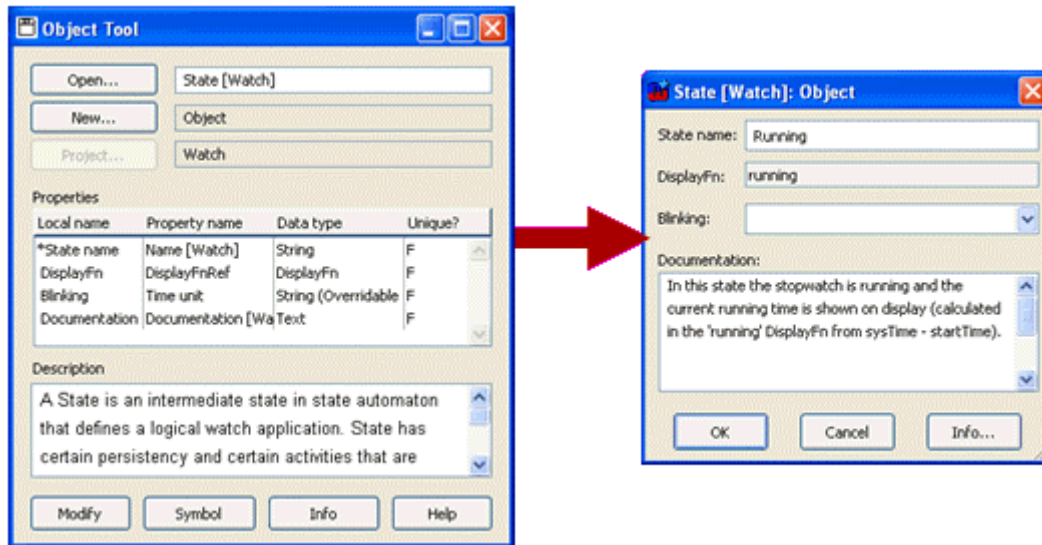
モデリング言語コンセプトを定義する事から言語の開発が始まります。生成するソースコード、製品のコンポーネントの用途、製品のアーキテクチャー、製品ラインの特徴等から各ドメインに依存したコンセプトが提起されます。MetaEdit+は、ドメインコンセプトをそのままモデリング言語として使用できるようにします（ドメインのコンセプトとルールを開発者に提供する環境）。モデル内でドメイン固有のコンセプトを使用することで、生産性が10倍向上し、絶大な費用対効果を挙げた例があります。（サクセスストーリーに、より多くの情報があります）

Define concepts

コンセプトを定義する

Use them in designs

コンセプトを使って設計する



MetaEdit+メタモデル作成ツールを使用することで、DSM 言語が迅速且つ簡単に開発できます。MetaEdit+を用いることで、ユーザー独自のドメインに適応したモデリング言語の構築という困難な作業が簡単に行えます。言語のコンセプトやそのプロパティとルールは、フォームを埋める形式又はグラフィカルに規定する形で設定できます。MetaEdit+には、6種類のメタモデル作成ツールと関連ツールが用意されています。

**オブジェクトツール**：モデリング言語の基本的なコンセプトを規定するためのツールです。オブジェクトは、例えば、処理、メッセージ、ボタン、状態等を、相互に接続可能で再利用可能なシンボルとして定義した、言語定義の中心的なエレメントです。

**プロパティツール**：他の言語コンセプトを特徴付けるプロパティ（属性、特性値）を定義するツールです。プロパティは、文字、文字列、数値、ブーリアン等の様々な型のデータとして定義できます。また、他のモデリング言語コンセプトや外部ファイルへのリンクとしても設定可能です。例えば、ステート名、機能のID、表示形式、データ型等が設定されます。

**リレーションシップツール**：継承、メッセージ、呼び出し、推移といったオブジェクト間の接続に対して、シンボルやプロパティを定義することで、リレーションシップを構築します。

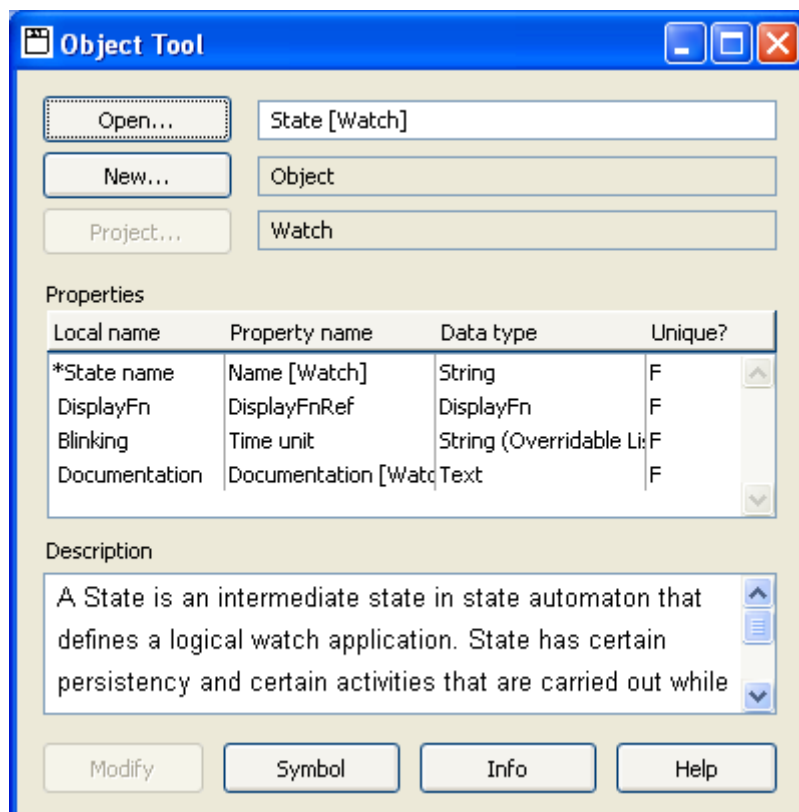
**ロールツール**：“継承リレーションシップのスーパークラス”、“メッセージリレーションシップのクライアント側”の様に、リレーションシップの始点と終点の役割を規定します。

**ポートツール**：他のオブジェクトと接続する際のインターフェースの制限を、“ハードウェアカードに対する入力はデジタル”や“コントロールフローはプロセスのエレメントの下部に接続”の様に定義できます。

**グラフツール**：ステート図やコンポーネント図の様な全てのモデリング手法を実現できます。モデリング手法は、他のツールで定義された、オブジェクト、リレーションシップ、ロールが相互に接続される条件を、バインディングやルールとして規定することで実現されます。エクスプロージョン、デコンポジッション及びモデリングコンセプトの再利用の機能も持っています。

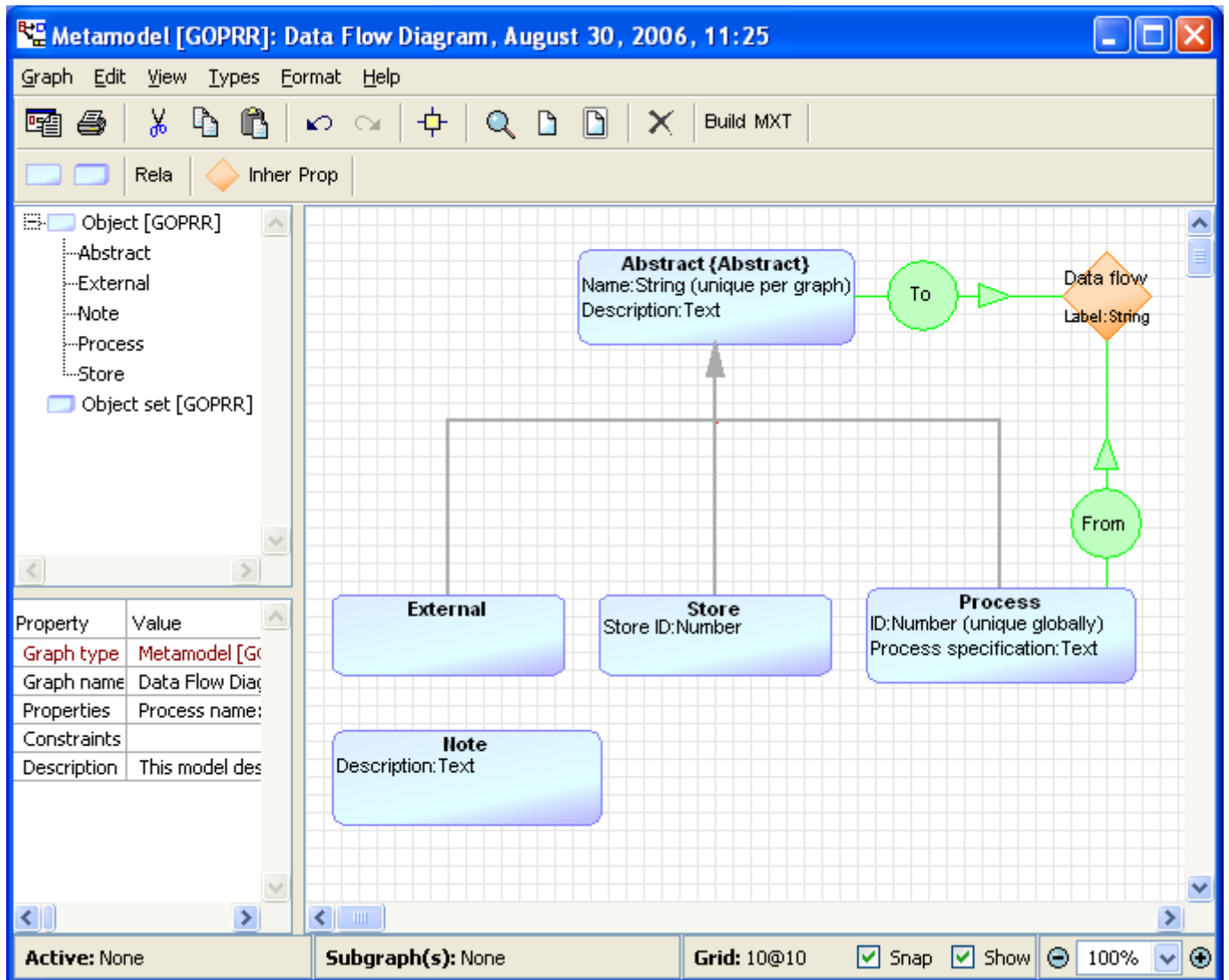
## 統合されたメタモデリングツール

MetaEdit+は、言語のコンセプト、言語のルール、言語の制約を、ジェネレータや表示と一緒に他の言語と結合する為の、統合されたメタモデリングツールを提供します。これらのテンプレートベースのメタモデリングツールでは、モデリングコンセプトの名前とプロパティを入力します。これらの言語の定義は、MetaEdit+で即座に使用でき、言語の進化させて行くことも可能です。ここでメタモデルを変更すれば、同時にモデルもアップデートされます。



## グラフィカルなメタモデリングツール

グラフィカルなメタモデリングは言語作成の初期段階で利用できます：言語コンセプトの鍵になる部分と関連したルールを網羅する基本的なメタモデルを設計します。メタモデルとしての言語設計は MetaEdit+ にインポート可能です。また、シンボル、制約、ジェネレータ等の追加や、ダイアログやツールバーを変更することで完全な DSM ソリューションに拡張可能です。

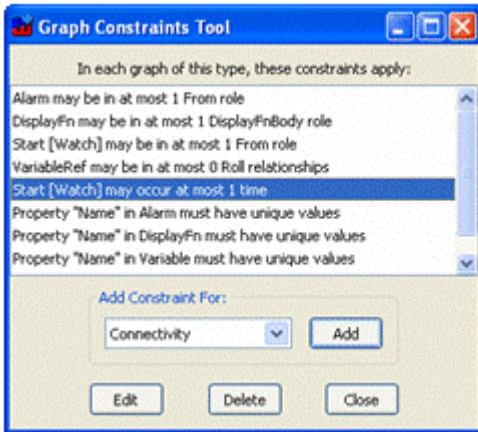


グラフィカルなメタモデリング環境は、モデルと分離して提供されます。その為、言語の進化には不向きです。実際のモデルを使って言語の修正をテストする方法にはうまく適合しません。また、表示、ジェネレータ定義、ダイアログやツールバーの修正も出来ません。MetaEdit+ Workbench には、テンプレートベースのメタモデリングツール、シンボルエディター、ジェネレータエディタとデバッガ、ダイアログエディタ等のツールがあり、言語設計の拡張や実装の全体に渡って有効です。

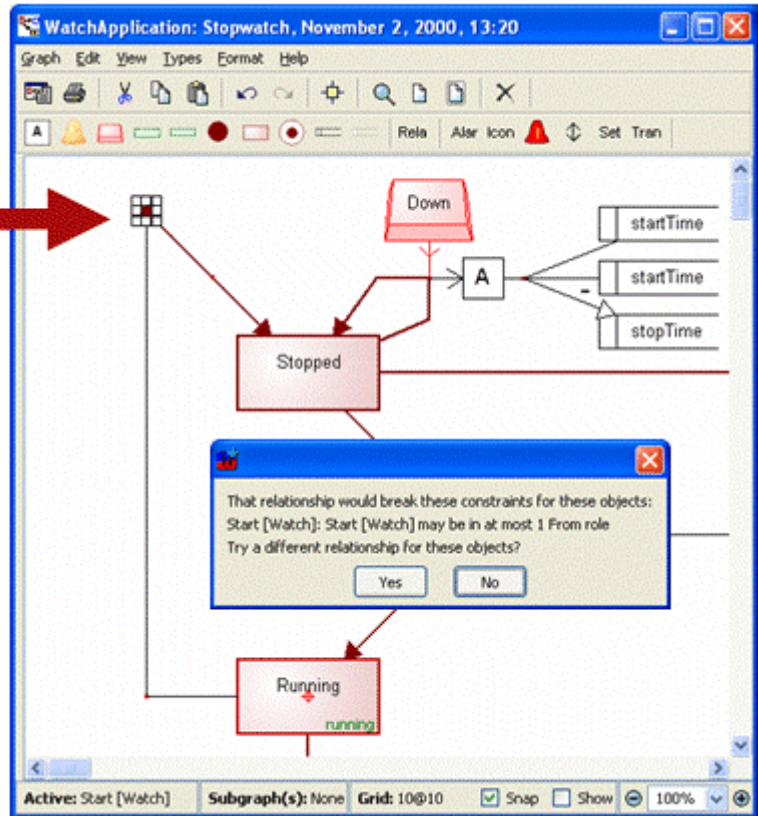
## ステップ2：ルールを選定する

ドメインには、部品定義と相互接続に関するたくさんのルールがあります。MetaEdit+のGOPRRメタモデル作成ツールを使用すれば、これらのルールを簡単に規定できます。開発者がモデリング言語を使用している時でも、ルールを変更することが可能です。MetaEdit+を使えば、開発者の環境を自動的に新しいドメインルールにアップデートできます。

ルールを選ぶ



ルールに従ってモデリング



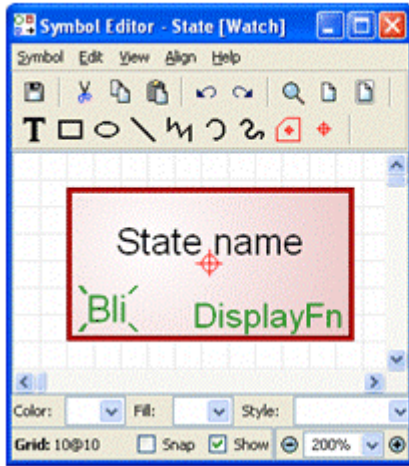
MetaEdit+では、モデル内及びモデル間のルールを規定できます。1つのモデリング言語では、そのコンセプトの相互関連性とオブジェクト間接続数が定義されます。例えば、“InitialState”のインスタンスはただ1つの“TriggeringConnection”を持ち、そのインスタンスは同じ通知イベントを最低でも2ヶ所に送る必要があるという具合です。こういったルールを定義することで、MetaEdit+を使っている開発者が、必ず正しい定義を行うことが保障されます。

エクスプロージョン、デコンポジッション、再利用といった多様なモデリングコンセプトも利用可能です。例えば“Productfeature”というドメインコンセプトは、1つ以上の振舞を示した状態図を持つことができるというものが、エクスプロージョンの例として挙げられます。

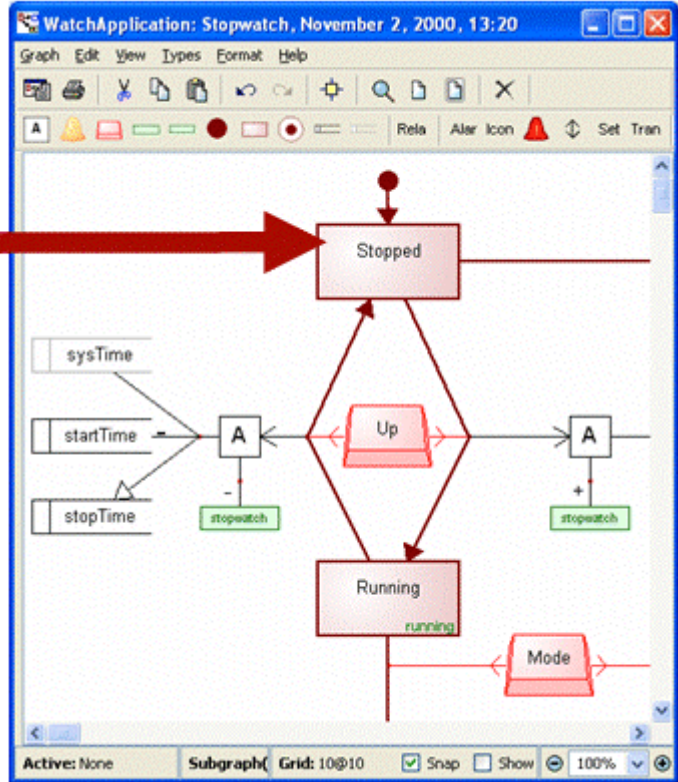
### ステップ3：シンボルの描画

言語のコンセプト（接続やそのルール）を表示させることができます。シンボルを描くだけで、全てのグラフィカルな振舞が自動的に描かれます。

シンボルを描く



シンボルを使ってモデルを描く



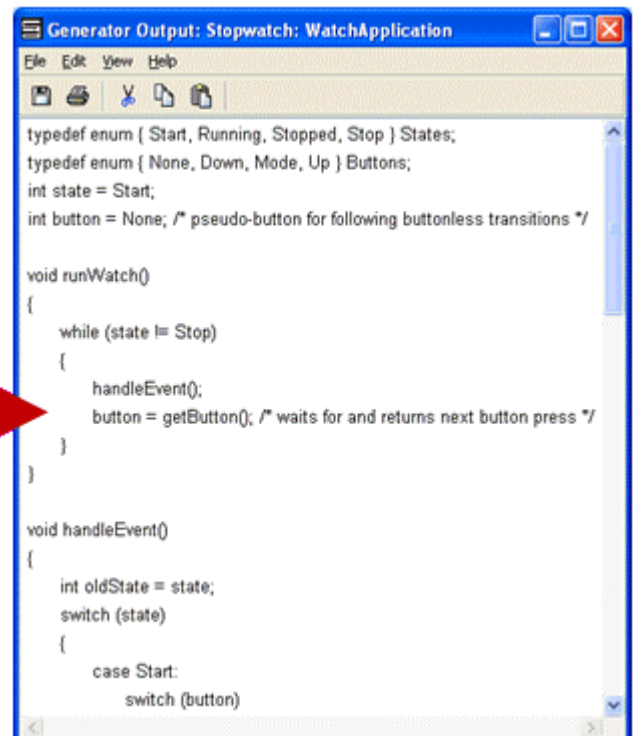
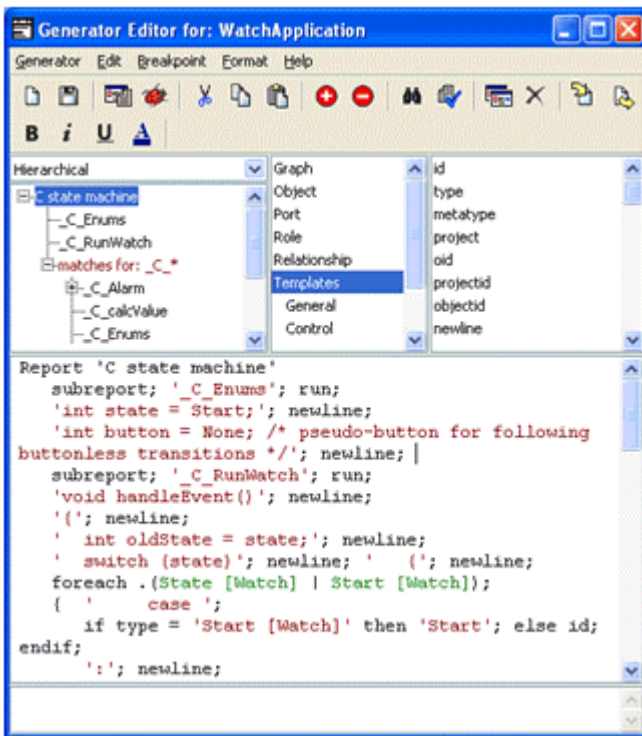
シンボルエディターを使えば、簡単にグラフィカルな表示を作成できます。シンボルエディターには、幾つかのシェイプと標準的なグラフィックツールで馴染みのあるツールがあります。それにより、多様なシェイプ、色、線の太さや形状、テキストの形状、配置が選択可能で、グラフィックファイルのインポートを行うことも出来ます。動的なグラフィックの振舞を実現するために、表示する値を計算したり、プロパティー値を条件にしてシンボルの要素を表示することが可能です。共通で使用するシンボルや部品は、再利用するためにシンボルライブラリーに保存可能です。

## ステップ4：ジェネレータの作成

ドメインを取り込んだモデルは色々な目的に利用できます。MetaEdit+を使うことで、モデル整合性のチェックをする為のレポートを作成、メトリックの生成、モデルの関連性解析、データ辞書の作成、ドキュメントの生成、コードや構成情報の生成及び、モデルの（シミュレータ、バージョン管理ツール等の）他のプログラムへのエクスポートが行えます。先進のスクリプトコマンドを使用して、設計を多彩なフォーマットで出力でき、複数のファイルへの出力が扱え、外部プログラムを呼び出せ、OSのコマンドを最大限利用できます。

ジェネレータを作成

ジェネレータを実行



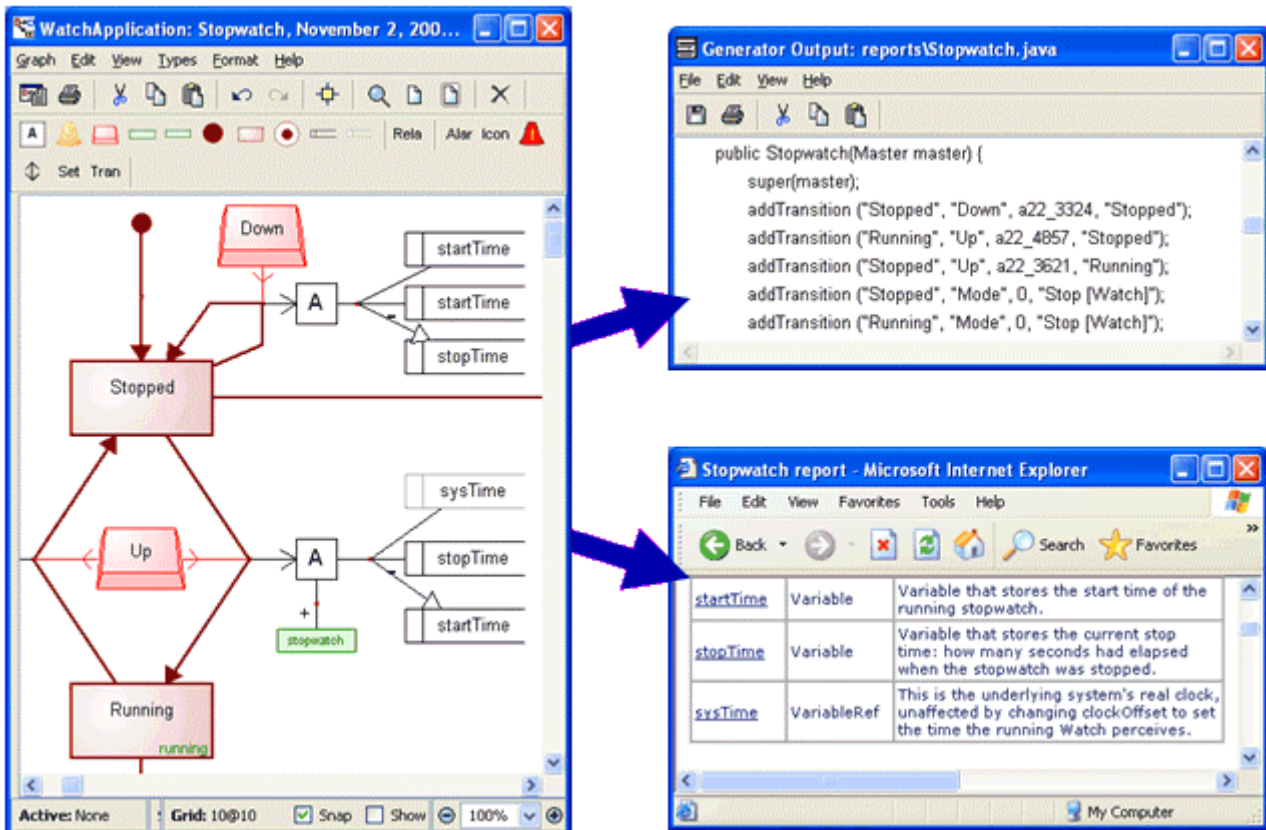
MetaEdit+のジェネレータエディタを使えば、レポートのカスタマイズとユーザー定義のレポート作成が可能です。ジェネレータエディタでは、レポートに出力したい設計データとその出力フォーマットを選択できます。様々な設計モデルをこのフォーマットに従って出力するために、このレポート定義が使われます。ジェネレータの動作を、ジェネレータデバッガを使ってテスト・デバッグ可能です。



## ステップ5：作製した言語を使う

MetaEdit+で作製された言語とジェネレータは、直ぐに開発に使えます。作製される開発ツールは独自のモデリング言語であり、望みどおりのコードを生成し、独自の基準に従ったドキュメントを生成します。それには現行の（構成データの読み込み、コンポーネントライブラリの参照、シミュレータの実行、外部ツールの実行等の）アプリケーション開発環境も統合されています。その他のツールからも MetaEdit+内のモデルにアクセスできます。また、他の環境との統合も可能です。MetaEdit+はただの描画ツールではありません。全ての主要なプラットフォームで動作する、純粋なマルチユーザー、マルチプロジェクトのモデリング環境です。

### モデリング言語を使って製品とドキュメントを生成する



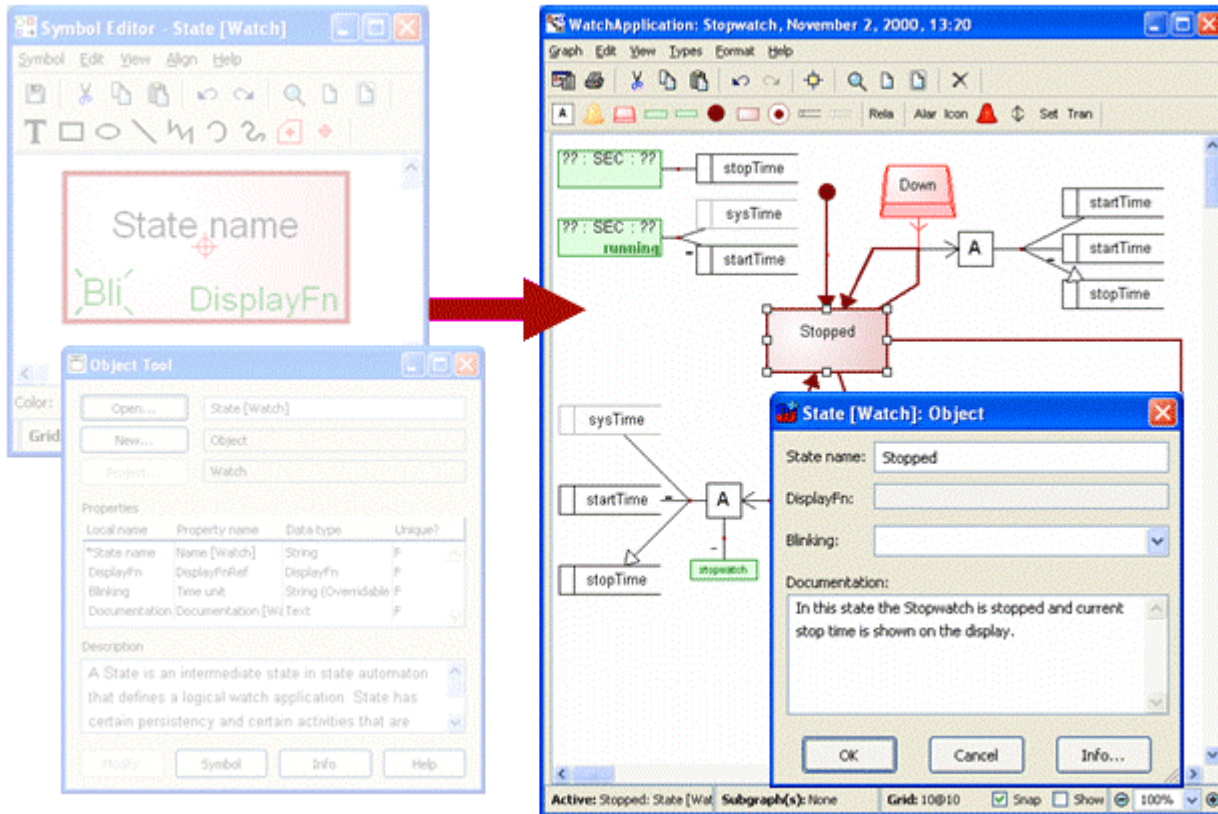
言語を使ってモデリングを行っている場合でも、モデリング言語の修正が可能です。MetaEdit+は、モデリング言語の修正が作成中のモデルに影響を与えない範囲で、作成中のモデルをアップデートします。モデリング言語修正の適用とモデルのアップデートは、MetaEdit+を使ってモデリングを行っている開発者に見えない状態で行われます。

## 《MetaEdit+ でモデリング言語を使用する》

MetaEdit+は、複数のユーザーと複数のプロジェクトに対して、全ての主要なプラットフォームで動作する完全なモデリングツールの機能を提供します。レポジトリから言語を取り出すことで、MetaEdit+ Workbenchで定義されたモデリング言語に自動的に従います。

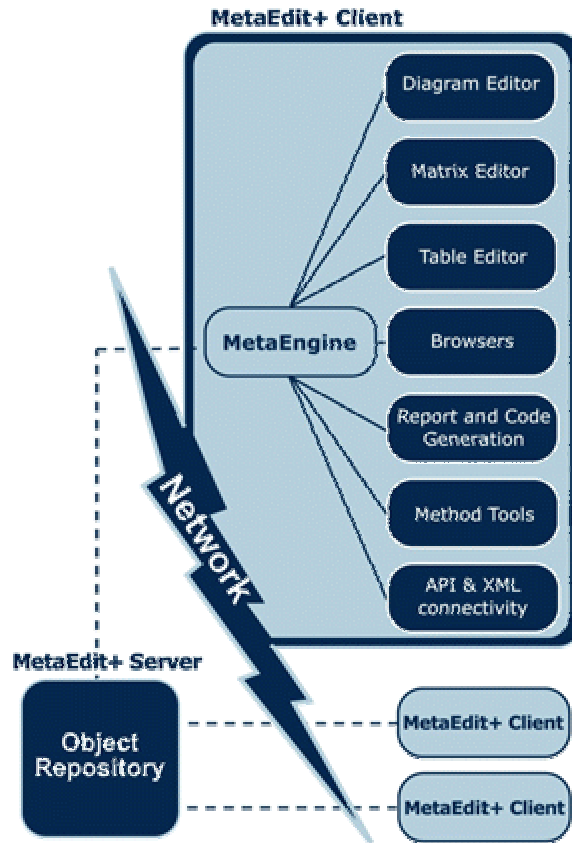
独自のモデリング言語を設計する

モデリング言語を使用する



MetaEdit+は独自の言語をサポートする完全なモデリングツールを提供します。チーム全体が即座に、必要性に合わせて切り替えられる、グラフィカルな図、マトリクス又は表を使った設計の編集を開始できます。また、フィルターを使ったデザインの表示、部品の追加、他の設計とのモデルのリンク、事前に或いはユーザーによって定義された多様なレポートを使ったモデルのチェックが行えます。DSM 言語内の設計を、ウェブやワードプロセッサに渡したり、製品の為の完全なソースコードを生成することが出来ます。自身で管理できる、欲しいツールを手にすることが出来ます。

## マルチツール、マルチユーザの環境



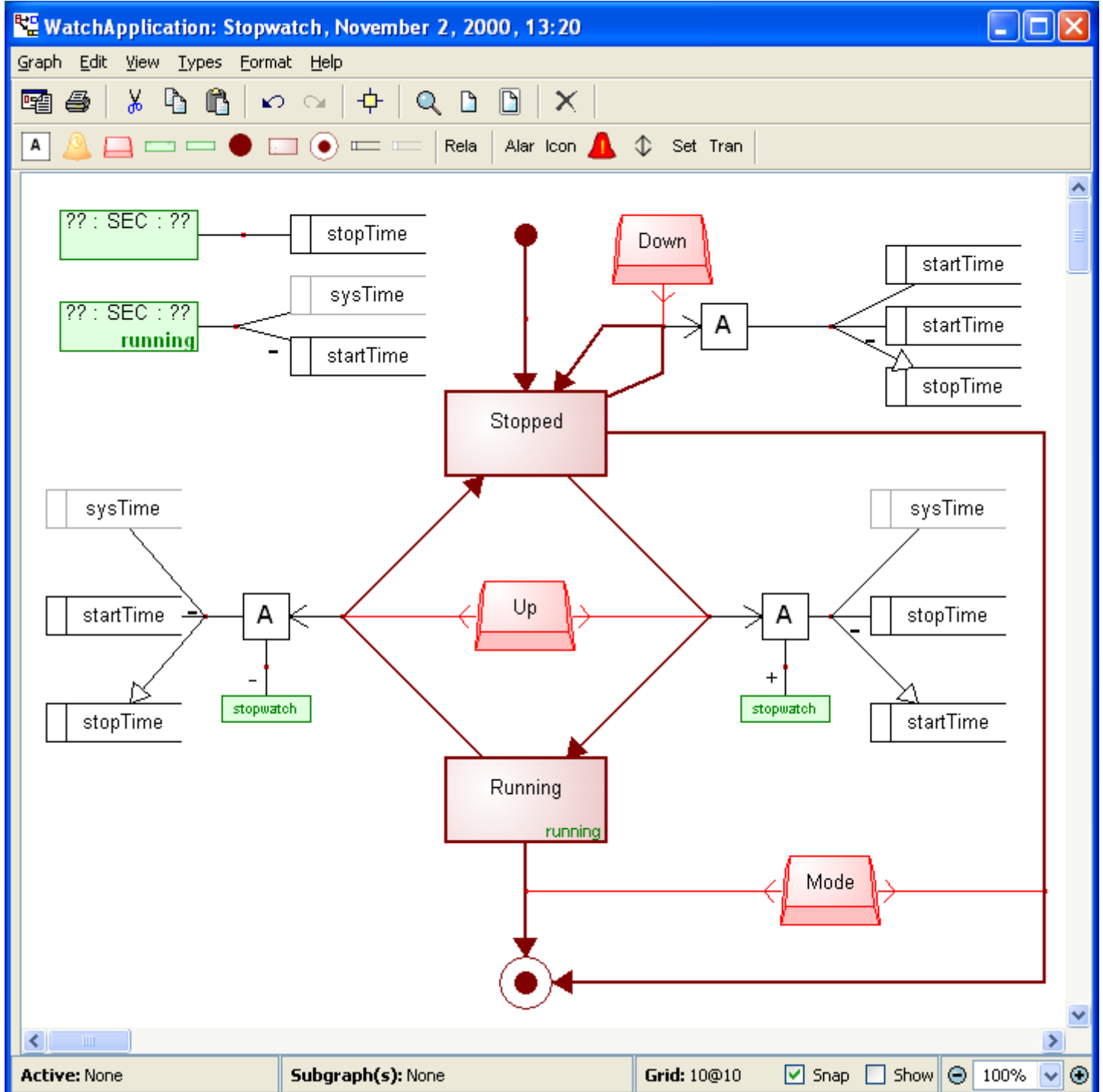
MetaEdit+は、作業に最適なツールを選択できる統合されたツール群を提案します。同じデータを、必要性に合わせて切り替え、グラフィカルな図、マトリクス又は表の各形式で簡単に見ることが出来ます。フィルターを使ったデザインの表示、部品の追加、他の設計とのモデルのリンク、事前に或いはユーザーによって定義された多様なレポートを使ったモデルのチェックが行えます。モデリングの結果は、ウェブやワードプロセッサに引き渡せ、製品の為の完全なソースコードを生成することも出来ます。

全てのツールは、ツール間で整合性を維持・管理されたオブジェクト・リポジトリを通して統合されています。MetaEdit+は、シングルユーザーのワークステーション環境でも、サーバーに接続された多数のワークステーションで同時に実行させる環境でも利用可能です。MetaEdit+は、この方法により完全な共同作業のソリューションを提案します。

図に示されている個々のツールを、以下で詳しく説明します。

## ダイアグラムを使った視覚的なモデリング

MetaEdit+では、グラフィカルダイアグラムを使ったモデリングが行えます。ダイアグラムエディターは、設計情報をダイアグラムとして、作成、管理、保守可能なツールです。グラフの視覚的なモデルリングと、エクスプロージョン、デコンポジッション、部品の再利用の機能を使った幾つかのグラフのリンクが可能です。(異なったフェーズやレベルでの)設計モデルの再利用と、ウェブブラウザ、イメージビューアー、テキストエディター等の他のツールへのエクスポートが可能です。



ダイアグラムエディターは、以下の色々な設計活動を補助する機能を持っています。

- リレーションシップの選択とロールの選択がコンテキストに反応することによるモデル作成の自動化
- モデリングコンセプトのツールバーからの選択
- コンポーネント選択ツールで、コンポーネントが表示、選択、再利用可能
- InfoTool を使って、設計情報間の接続を調査・ナビゲート可能
- マトリクスやテーブルをダイアグラムとして表示

マウスクリックで表示されるプロパティダイアログで、全ての設計情報にアクセス可能  
事前定義のレポート機能やチェック機能をカスタマイズしたレポートを使ってモデルのチェックが可能

ドキュメント生成機能で、モデルをウェブやワードプロセッサに出力可能

C++, Smalltalk, Java, CORBA IDL, Delphi, SQL 及びその他のユーザー定義のプログラム言語でコード生成可能

モデリングテクニック、モデリング部品、レポート、その他のテクニックへのリンクに関するヘルプテキストを表示可能

ダイアグラムエディターは、ダイアグラムの表示とフォーマットの為の全ての機能を持っています。

シンボルの縮尺を変更する

シンボルのオーバーラップ表示と透過表示

グリッドの設定とグリッドへの吸着

ダイアグラムのズーム表示

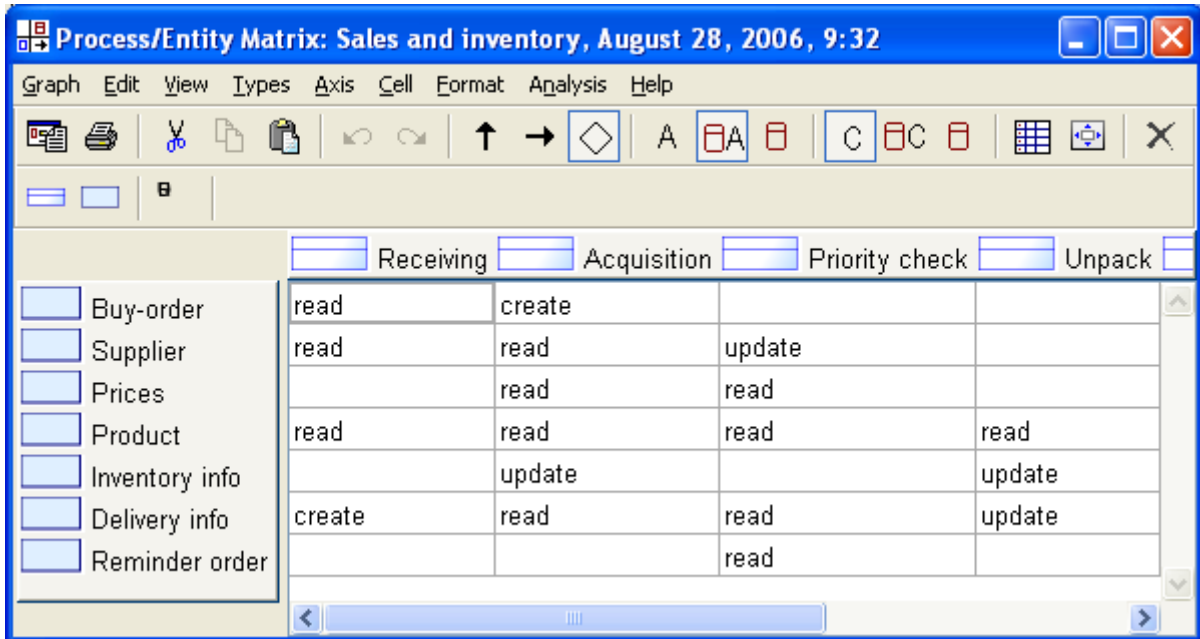
クリップボード、GIF、PICT へのダイアグラムのエクスポート

様々な形式でのダイアグラムの印刷

ハイパーテキスト機能を使ったオンラインヘルプ

## マトリクスを使った視覚的なモデリング

MetaEdit+はマトリクスを使ったモデリングをサポートしています。マトリクスエディターは、設計情報を“オブジェクトを持った2つの軸”と“リレーションシップを表すセル”で構成されたマトリクスとして扱います。このエディターでは、マトリクスの対角化、並べ替え、サブシステムへの分解、多様な表示オプションと言ったマトリクス操作の全ての機能が、標準的なオブジェクトやリレーションシップの作成・編集と同様に使えます。



マトリクスエディターは、以下の色々な設計活動を補助する機能を持っています。

- モデリングコンセプトを選択するための標準ツールバー

- リレーションシップの選択とロールの選択がコンテキストに反応することによるモデル作成の自動化

- コンポーネント選択ツールで、コンポーネントが表示、選択、再利用可能

- InfoTool を使って、設計情報間の接続を調査・ナビゲート可能

- ダイアグラムやテーブルをマトリクスとして表示

- マトリクスは対角化可能

- 軸やセルにあるマトリクスや部品を他のグラフに分解・結合可能

- マウスクリックで表示されるプロパティダイアログで、全ての設計情報にアクセス可能

- 事前定義のレポート機能やチェック機能をカスタマイズしたレポートを使ってモデルのチェックが可能

- ドキュメント生成機能で、モデルをウェブやワードプロセッサに出力可能

- C++, Smalltalk, Java, CORBA IDL, Delphi, SQL 及びその他のユーザー定義のプログラム言語でコード生成可能

- モデリングテクニック、モデリング部品、レポート、その他のテクニックへのリンクに関するヘルプテキストを表示可能

- ハイパーテキスト機能を使ったオンラインヘルプ

マトリクスエディターは、マトリクスを表示、フォーマットする為の以下の機能を持っています。

- マトリクス内の部品の表示と非表示

- マトリクスの書式の自動化

- 複数フォントのサポート

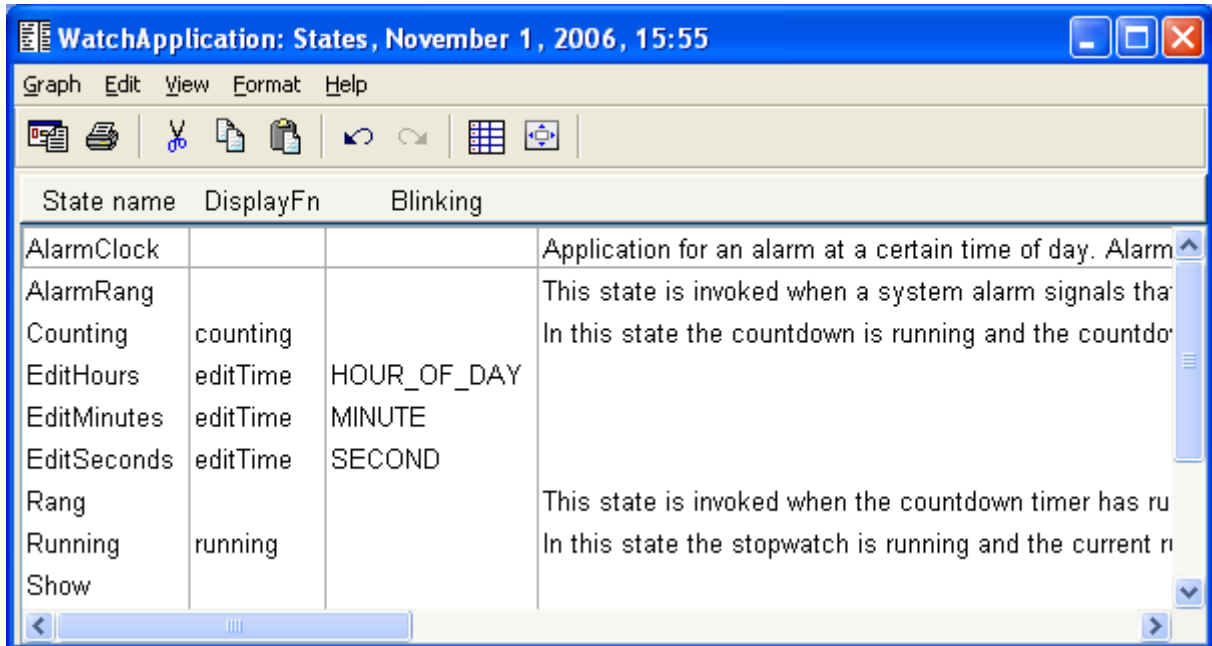
- マトリクス内のシンボルや文字仕様の表示と非表示

- マトリクスの ASCII (タブ区切り: 表計算ソフトに取り込む) と HTML へのエクスポート

- マトリクスの印刷

## テーブル、フォーム、テキストベースのモデリング

グラフィカルダイアグラムとマトリクスに加えて、MetaEdit+はテーブル、フォーム、テキストによるモデリングもサポートします。テーブルエディターは、設計情報を表形式やフォームベースで表示するためのツールです。このツールを使えばオブジェクトをコンパクトなフォームで見ることができ、開発の初期段階に於いて特に便利に使えます。



テーブルエディターは、以下の色々な設計活動を補助する機能を持っています。

- コンポーネント選択ツールで、コンポーネントが表示、選択、再利用可能

- InfoTool を使って、設計情報間の接続を調査・ナビゲート可能

- ダイアグラムやマトリクスのオブジェクトをテーブルとして表示可能

- テーブルにリストされたオブジェクトは、様々なオプションで保存可能

- テーブル内のオブジェクトは、他のグラフに結合、分解可能

- マウスクリックで表示されるプロパティダイアログで、全ての設計情報にアクセス可能

- 事前定義のレポート機能やチェック機能のカスタマイズしたレポートを使ってモデルのチェックが可能

- ドキュメント生成機能で、モデルをウェブやワードプロセッサに出力可能

- C++, Smalltalk, Java, CORBA IDL, Delphi, SQL 及びその他のユーザー定義のプログラム言語でコード生成可能

- モデリングテクニック、モデリング部品、レポート、その他のテクニックへのリンクに関するヘルプテキストを表示可能

テーブルエディターは、以下のテーブルのフォーマットと表示機能を持っています。

- 選択したカラムを表示・非表示

- カラム幅の自動設定

- 複数フォントのサポート

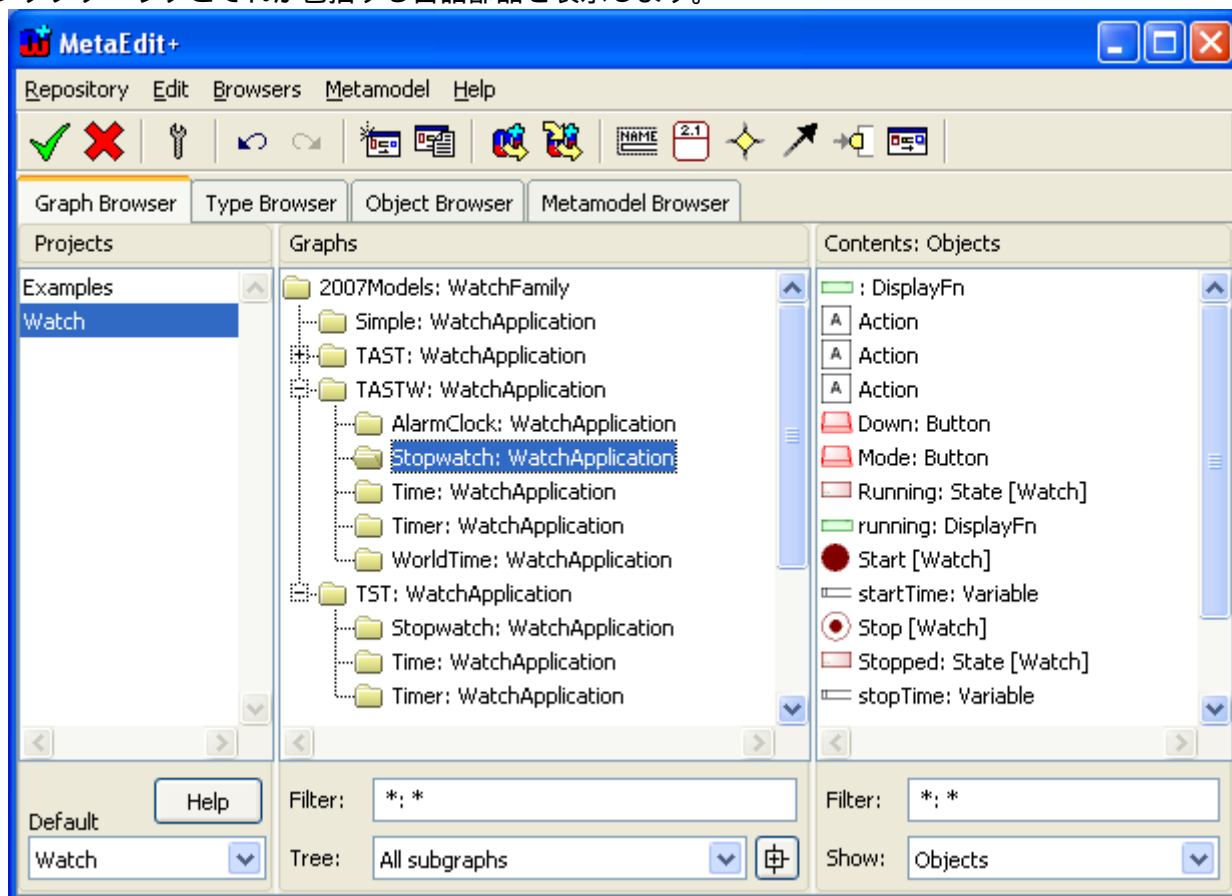
- テーブルの ASCII (タブ区切り: 表計算ソフトに取り込む) と HTML へのエクスポート

- テーブルの印刷

- ハイパーテキスト機能を使ったオンラインヘルプ

## ブラウザー

MetaEdit+には、リポジトリ内容の表示と編集のための幾つかのブラウザーがあります。タイプブラウザーは、モデリング言語と、グラフとその部品の両方に使えます。グラフブラウザーは、グラフの階層ベース（結合、分解の処理を選択）の表示を行い、各々のグラフの設計部品にアクセスできます。メタモデルブラウザーは、モデリングテクニックとそれが包括する言語部品を表示します。



ブラウザーで以下の機能が実行できます。

プロジェクトの作成と管理

複数のプロジェクトの設計や言語情報（メタモデル）の表示

言語のタイプを基準にした設計情報とグラフの表示

フィルターを使った設計情報とグラフの選択表示

エディターを使ってグラフを開く

InfoTool を使った、設計情報間の接続を調査・ナビゲート

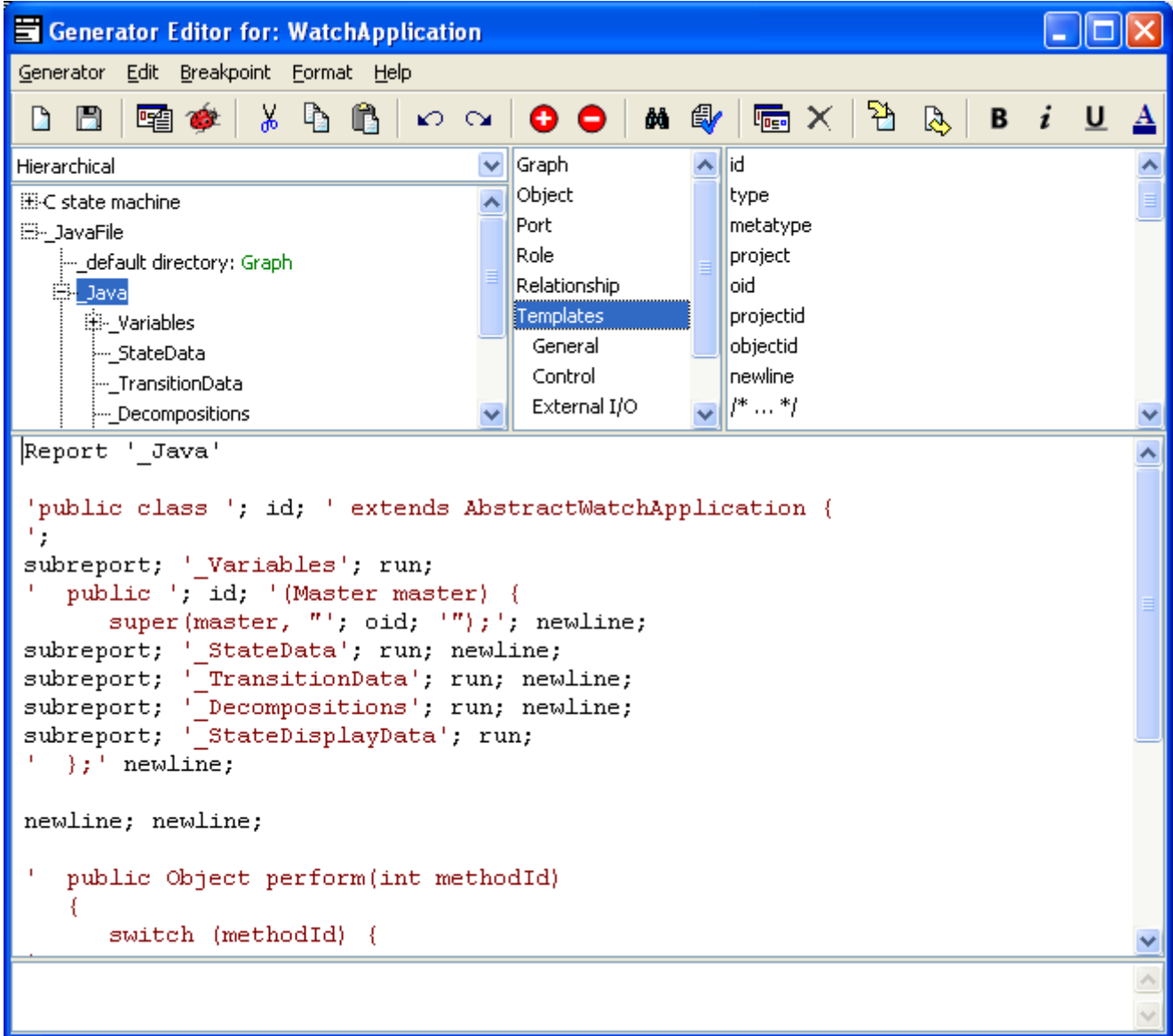
レポートの作成と実行によるチェック、ドキュメント生成、コード生成

プロパティダイアログを使った、全ての設計情報へのアクセス



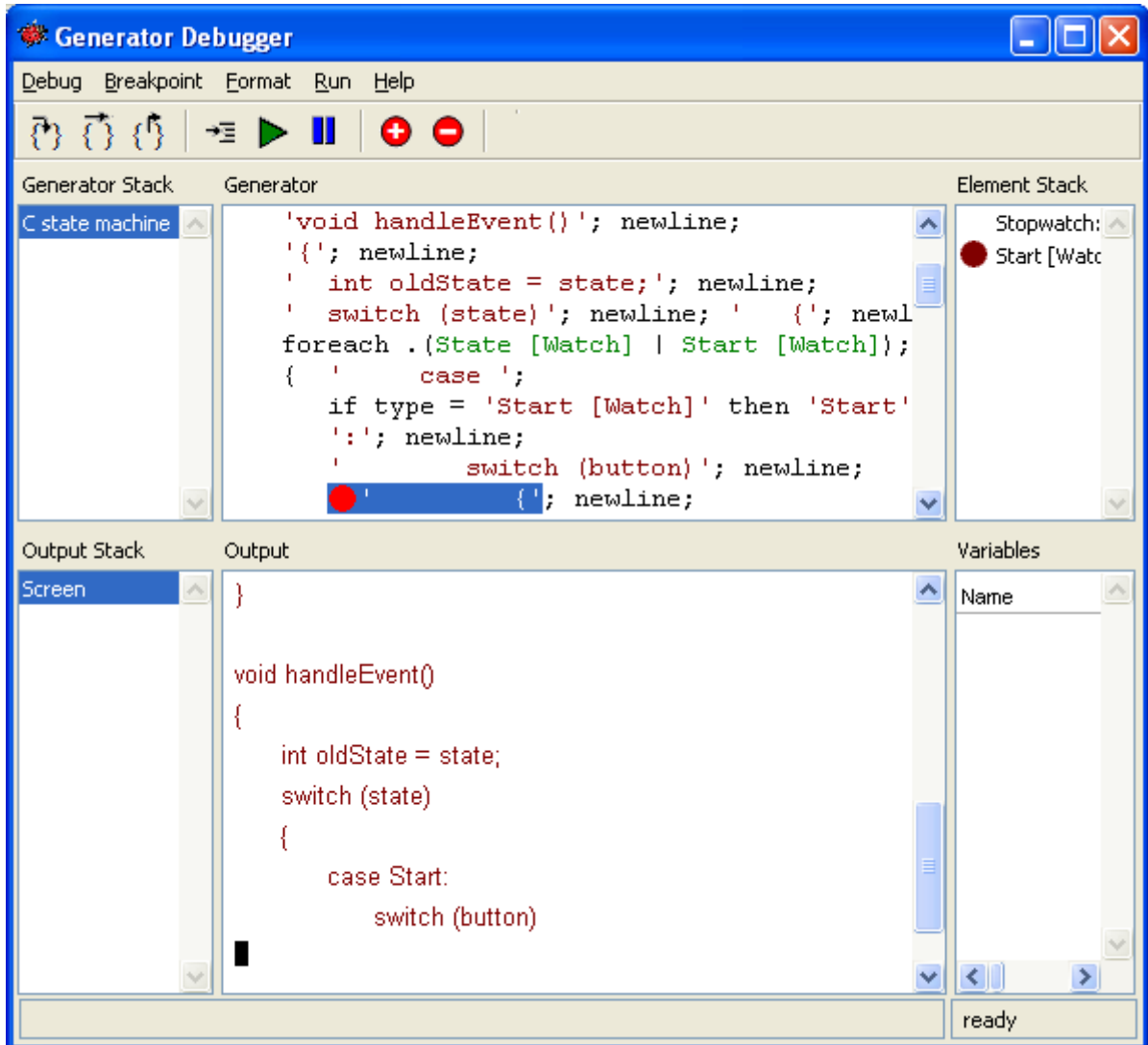
## コード、ドキュメント、メトリクス及びその他の出力の生成

コードジェネレータとその他のレポートは MetaEdit+ のジェネレータエディターを使って定義します。ジェネレータ定義言語を使って、モデルの渡り歩き方、モデルの値の出力方法、固定テキストの出力方法を規定します。コード生成、設定情報生成、ドキュメントやデータ辞書の生成、モデルの正当性のチェック、メトリクスの生成、モデルのリンク関係の解析、シミュレータ・コンパイラ・エミュレータ等へのモデルのエクスポートを行うジェネレータを作成可能です。



## ジェネレータデバッガ

ジェネレータデバッガは、ジェネレータの実行を調査、制御、追跡するためのツールです。ジェネレータデバッガによって、アクセスされた部品、使用された変数、生成されたコードと共にジェネレータの実行状態が1つのツール内に表示されます。

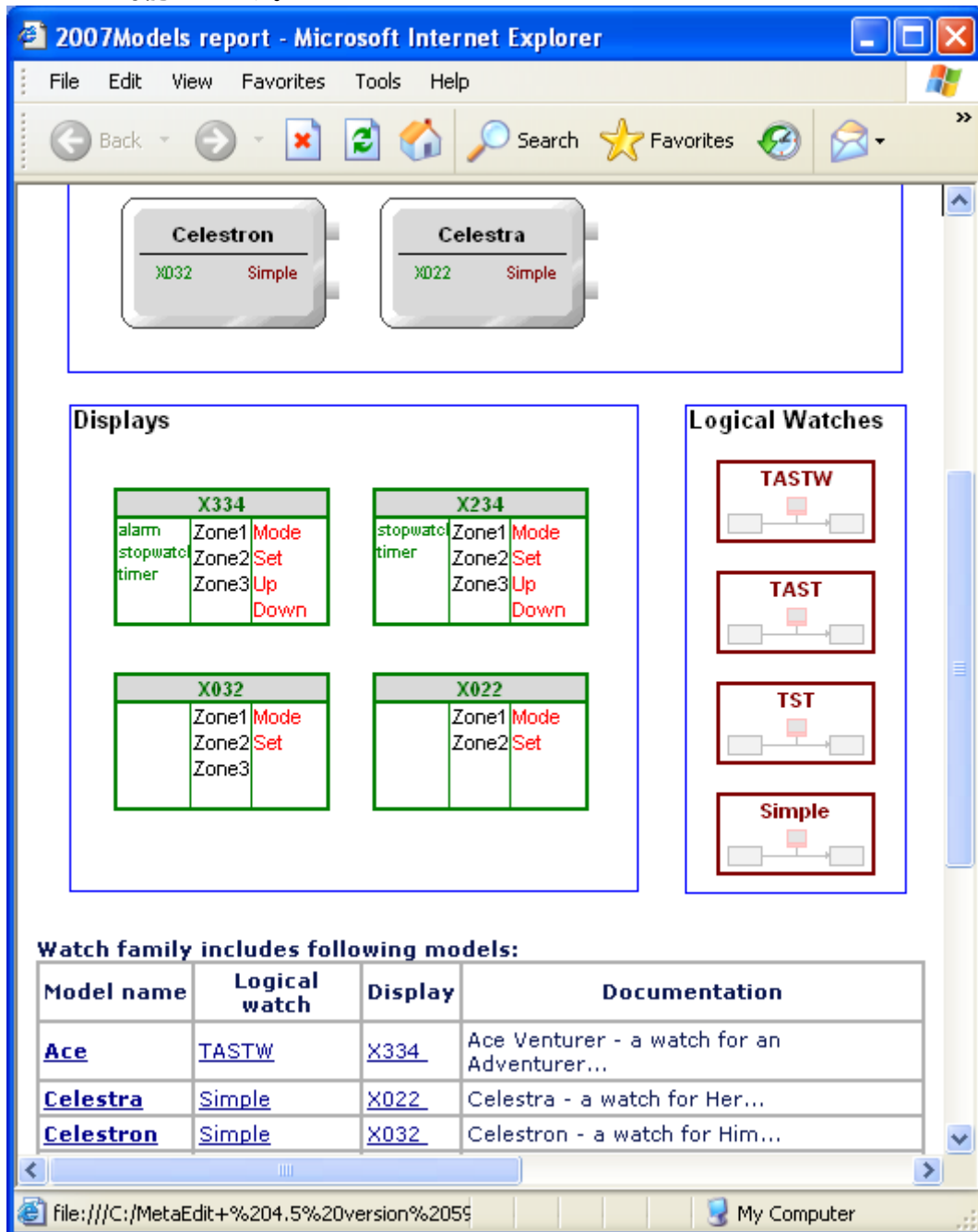


MetaEdit+には全てのグラフに対する事前に定義されたジェネレータがあります。これら全てのジェネレータは、ジェネレータエディタによって実行されます。これらのジェネレータには、ジェネレータエディタを使ってアクセスでき、独自の言語を作成する過程で変更・再利用が可能です。以下の事前定義のジェネレータとレポートがあります。

- コードジェネレータ
- ドキュメントレポート
- 標準レポート

## ウェブへのプロジェクトドキュメントの出力

全てのプロジェクト設計に関するドキュメントは、何時でも自動的にウェブ形式で出力可能です。生成されたHTML ファイルは、ダイアグラムと設計間のハイパーリンクを含んでおり、MetaEdit+を使わないウェブブラウザによるレビューを可能にします。



プロジェクトドキュメント生成では、プロジェクト全体の“ホームページ”とプロジェクト内にある各々のグラフの為にページを生成します。プロジェクトドキュメント生成は、上の Watch family ドキュメント生成のように、特定用途の為にカスタマイズできます。事前定義のプロジェクトホームページは以下の情報を含みます。

- プロジェクトの管理者
- プロジェクトの構成要員 ( e-mail 情報や連絡先情報を含みます )
- プロジェクトのステータス
- プロジェクトの構造図

簡潔なドキュメントと他のグラフのページへのリンクを含んだグラフとサブプロジェクト

各サブプロジェクトのページは同様の情報を含み、各グラフのページはダイアグラム・オブジェクト、それらのプロパティ、リレーションシップ、他のグラフへのリンクを含みます。

## Word へのプロジェクトドキュメントの出力

全てのプロジェクト設計に関するドキュメントは、何時でも、DOC ファイルとして自動的に生成できます。生成された DOC ファイルは、設計部品間のハイパーリンクとダイアグラムを含んでおり、テキストは修正可能です。

**Ball Game requirements: Use Case Diagram**

Status: draft  
Personnel: Joe Smith  
Documentation: This model describes the use cases for the ball game application.

Diagram picture: Ball Game requirements

**Graph dictionary**

Object	Type of Object	Documentation
<a href="#">Player</a>	Actor [UML]	Player who plays the ball game or views the high scores.
<a href="#">Ball game</a>	System [UML]	
<a href="#">Play a game</a>	Use case [UML]	The user can play a game which a traditional ball game: player moves the paddle and tries to remove the bricks from the field by hitting them with the ball. The player has three balls per game.

プロジェクトのドキュメント生成は以下の情報を含みます。

ドキュメントの情報を持った表紙

目次

プロジェクトの名前と詳細

プロジェクトの管理者

プロジェクトの構成要員と連絡先の詳細

プロジェクトのステータス

プロジェクトの構造図

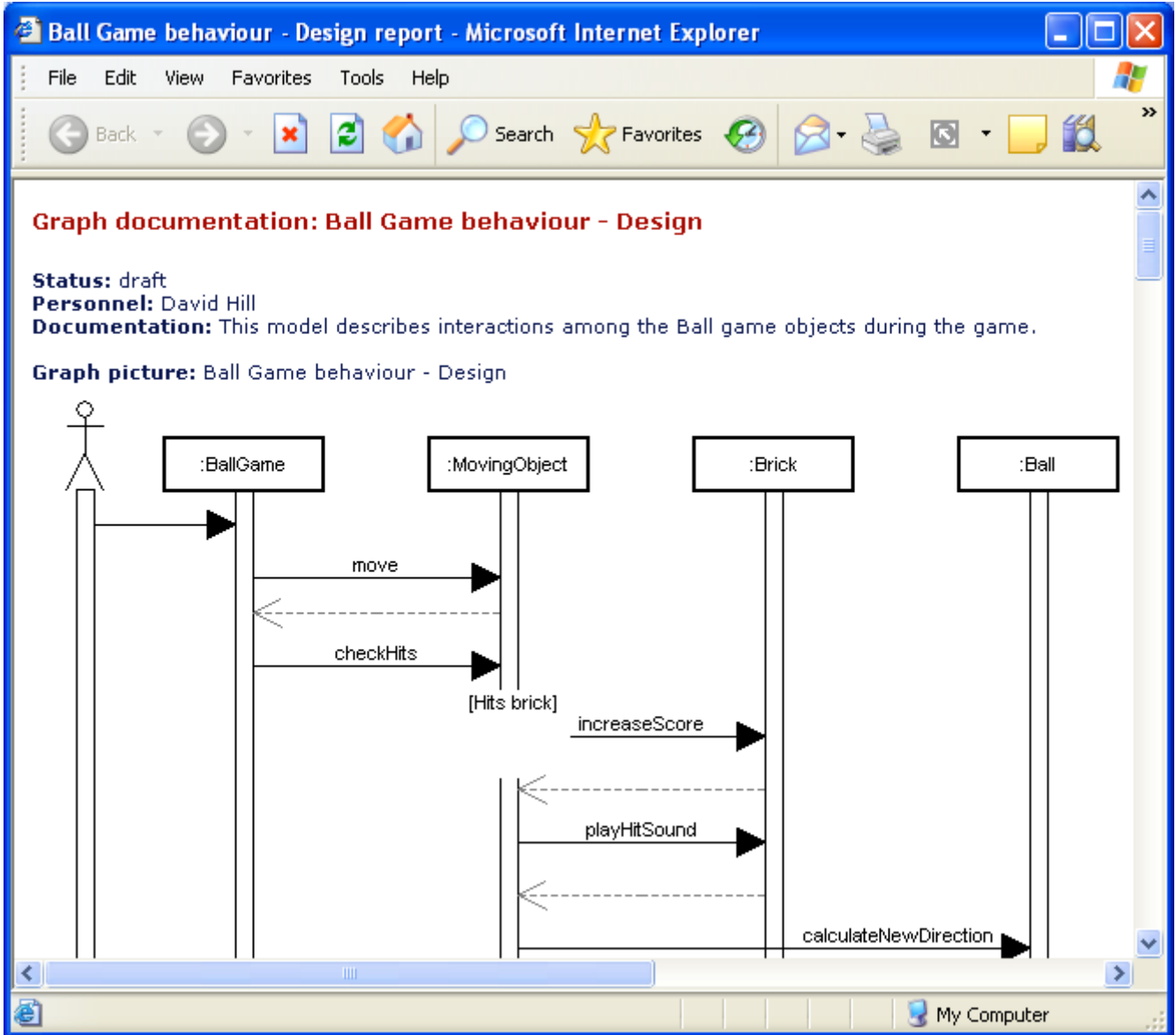
簡潔なドキュメントや各々のグラフの完全なドキュメントへのリンクを持った、グラフやサブプロジェクトの辞書

各サブプロジェクトのドキュメントは同様の情報を含み、各グラフのドキュメントはダイアグラム・オブジェクト、それらのプロパティ、リレーションシップ、他のグラフへのリンクを含みます。生成された DOC ファイルは、設計部品間で参照するためのハイパーリンクも含みます。

DOC ファイルは MetaEdit+ のスタイルテンプレートを使用しています。独自のテンプレートを使い、希望スタイルのドキュメントを生成するように修正することも可能です。MetaEdit+ の全てのレポートに於いて、カスタマイズが可能です。(何をレポートするか? どのようなフォーマットにするか?)

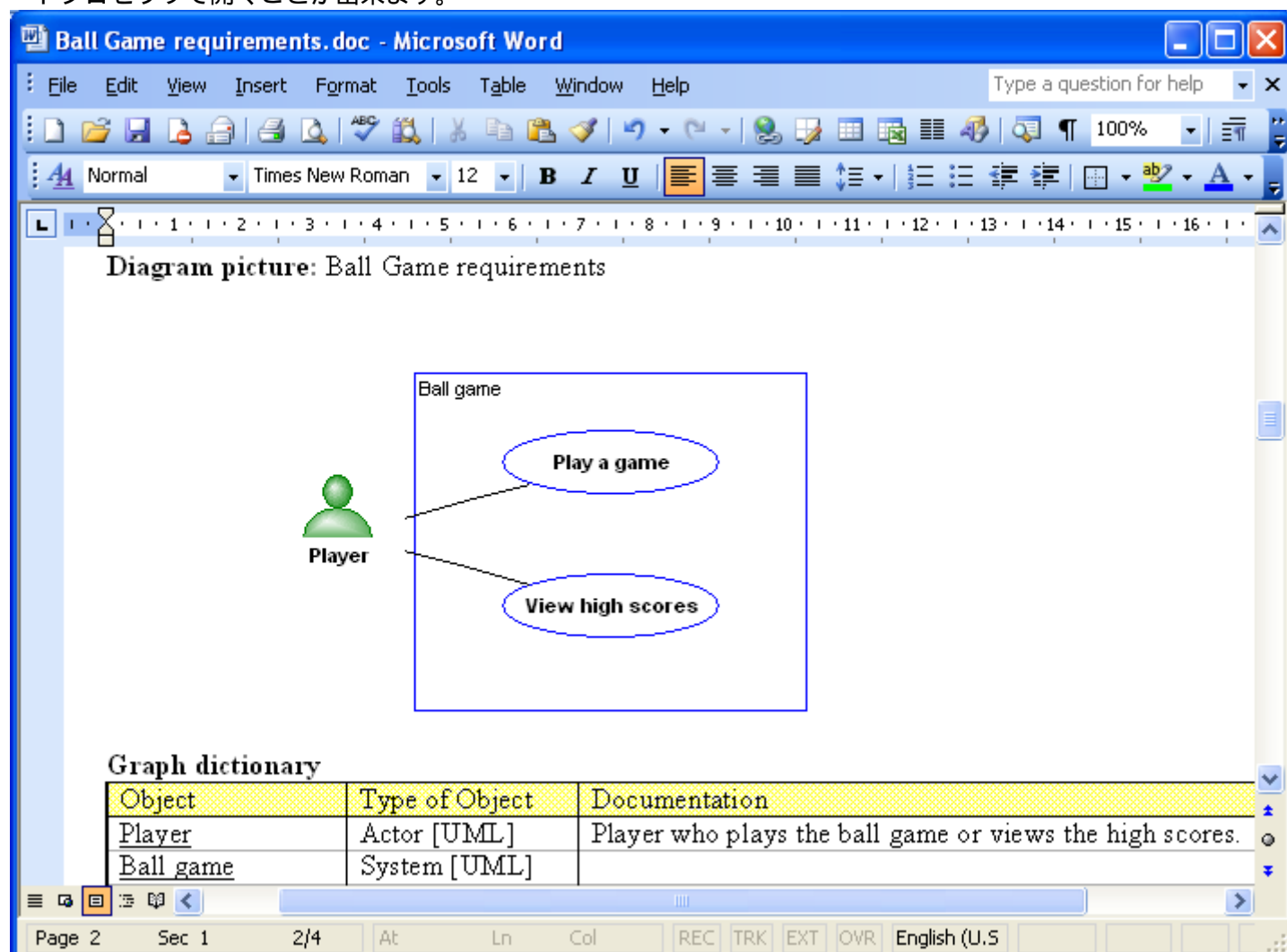
## ウェブへのグラフドキュメントの生成

マウスクリックで、全てのグラフに関するドキュメントが自動的にウェブに生成されます。生成される HTML ファイルは、PNG 形式のダイアグラム、設計部品のドキュメントとプロパティ、及び（オブジェクトの連携、サブグラフといった）それらのハイパーリンクを含みます。生成されたドキュメントを使えば、MetaEdit+ を使わずにウェブブラウザでレビューを行えます。



## Word へのグラフドキュメントの出力

モデルを自動的に RTF フォーマットのドキュメントで出力します。生成された RTF ファイルはダイアグラムと (MetaEdit+ ないの部品に対応した) 設計部品間のハイパーリンクを含みます。ドキュメントファイルはワードプロセッサで開くことができます。



The screenshot shows a Microsoft Word window titled "Ball Game requirements.doc". The main content area displays a diagram with a green actor icon labeled "Player" connected to two ovals within a box labeled "Ball game": "Play a game" and "View high scores". Below the diagram is a "Graph dictionary" table.

Object	Type of Object	Documentation
<u>Player</u>	Actor [UML]	Player who plays the ball game or views the high scores.
<u>Ball game</u>	System [UML]	

RTF フォーマットのグラフドキュメントは以下の情報を含みます。

- 全てのグラフに対する情報

- 詳細

- ステータス

- 作成者

- PNG フォーマットのダイアグラム

- ダイアグラム内の各々のオブジェクトから詳細な内容へのリンクの辞書

- 各々の設計オブジェクトに関する情報

- ドキュメント

- プロパティの詳細

- 他の関連オブジェクトへのハイパーリンク

- サブグラフへのハイパーリンク (分解 / 結合)

RTF ファイルは MetaEdit+ のスタイルテンプレートを使用しています。独自のテンプレートを使い、希望スタイルのドキュメントを生成するように修正することも可能です。MetaEdit+ の全てのレポートに於いて、カスタマイズが可能です。(何をレポートするか? どのようなフォーマットにするか?)

## 《MetaEdit+のその他の機能》

### 事前定義のモデリング言語サポート

MetaEdit+は、モデル解析ツール、コードジェネレータ、ドキュメントジェネレータなどを一体とした、数百ものDSM言語を構築する為に利用されてきました(see [application domain examples](#))。多くの統一モデリング言語がMetaEdit+で提供されています。これら言語は、直ぐに作業を始められるように、再利用可能なモデリング言語部品のライブラリとして提供されています。

Object-Oriented Modeling Languages	
Booch et al.	<a href="#">Unified Modeling Language</a>
Rumbaugh	<a href="#">Object Modeling Technique</a>
Booch	<a href="#">Object-Oriented Design</a>
Coad/Yourdon	<a href="#">Object-Oriented Analysis and Design</a>
Coleman et al.	<a href="#">Fusion</a>
Henderson-Sellers	<a href="#">Moses</a>
Embley et al.	<a href="#">Object-Oriented Systems Analysis</a>
Schlaer/Mellor	<a href="#">OODLE</a>
Structured Modeling Languages	
Yourdon	<a href="#">Structured Analysis and Design</a>
Ward/Mellor	<a href="#">Real-Time SA/SD</a>
Business Modeling Languages	
IBM	<a href="#">Business Systems Planning</a>
Porter	<a href="#">Value Chains &amp; Value Systems</a>
Goldkuhl	<a href="#">Activity Analysis</a>

様々なモデリング言語を MetaEdit+で使用すれば、特定のモデリング言語に強く依存した部分以外（ツールを使う上でのユーザーインターフェース等）が統一されるため、操作が簡単になります。1 つのツールの使用方法のみを学習すればよくなる為、様々なモデリング言語を使用する場合の習熟時間を短縮できます。又、異なったモデリング言語の為に複数のモデリングツールを購入する必要がなくなります。これによって、トレーニング等にかかる費用と人的資源を節約できます。その上、MetaEdit+は、複数のモデリング言語を同時にサポートすることで、異なる言語とパラダイム間の移行方法を提供可能です。同様に、複数言語のサポート機能により、異なったモデリング言語を通してデータを再利用・連携可能です。これにより、異なったモデリング言語間の情報の流れを維持できます。

## コードジェネレータ

MetaEdit+は、事前定義又はユーザー定義のプログラム言語に対する自動コード生成をサポートしています。コードを自動生成できる可能性は、使用しているモデリング言語とターゲットプログラム言語に依存します。以下のプログラム言語に対応した事前定義のコードジェネレータがあります。

Smalltalk  
C++  
Java  
Delphi (Object Pascal)  
SQL  
CORBA IDL

これらのカスタマイズ可能なコードジェネレータは、ジェネレータエディタによって実装されます。ジェネレータエディタを使ってこれらのコードジェネレータにアクセス可能で、独自のコードジェネレータを作成するためにそれらを変更・再利用できます。

## 自動ドキュメント生成

MetaEdit+は、クリック 1 つでウェブやワードプロセッサのフォーマットのドキュメントを生成するドキュメント化レポートの機能を提供します。MetaEdit+はモデルを解析・チェックする為のビルド前のレポート、及び Word・RTF・HTML・XML・XMI を使ったドキュメント作製機能を提供します。手作業によるドキュメント作製作業や旧態依然としたのドキュメント作製はもはや不要です。

以下形式でプロジェクトのドキュメントを作成します。

**Export to HTML** : 全てのプロジェクトを HTML 形式で出力します。( 画像は GIF フォーマットになります )

**Export to Word** : 全てのプロジェクトを Word 形式で出力します。( 画像はベクターフォーマットになります )

以下の形式でグラフモデルのドキュメントを作成します。

**Export graph to HTML** : グラフィックモデルを HTML 形式で出力します。( 画像は GIF フォーマットになります )

**Export graph to Word** : グラフィックモデルを Word 形式で出力します。( 画像はベクターフォーマットになります )



## 他の環境との統合

MetaEdit+を使えば、他のツールとの洗練された統合を実現できます。新しいツール統合の試みです。

- 1 . API を通じた MetaEdit+の機能とモデルデータへのプログラムに基づいたアクセス
- 2 . XML フォーマットでのモデルのインポートとエクスポート
- 3 . MetaEdit+を自動動作させる為のコマンドラインパラメータ
- 4 . ジェネレータからの外部コマンド実行

## プログラムに基づいたアクセスを実現する為のAPI

API はモデルエレメントの読み込み、作製、アップデートの為のインターフェースを提供します。同様に、スクリプト動作やシミュレーションをサポートするために MetaEdit+を制御することも可能です。このリアルタイムの動的リンクは多くの用途を提供します。

シミュレーション：コードが実行されている間の MetaEdit+内でのモデルのアニメーション表示（例を参照）

モデルの変換：全てのモデルに同じ変更を施す

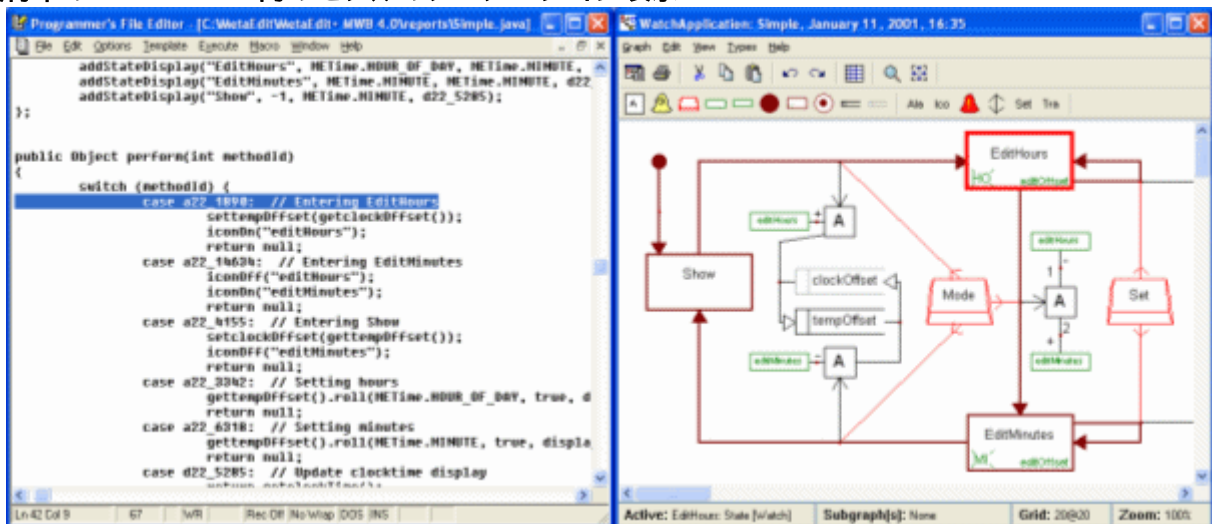
差分解析：モデルとデータベースを比較する（例を参照）

ビルド：多くのデータベースを使った無人コード生成

モデルのチェックと手引き

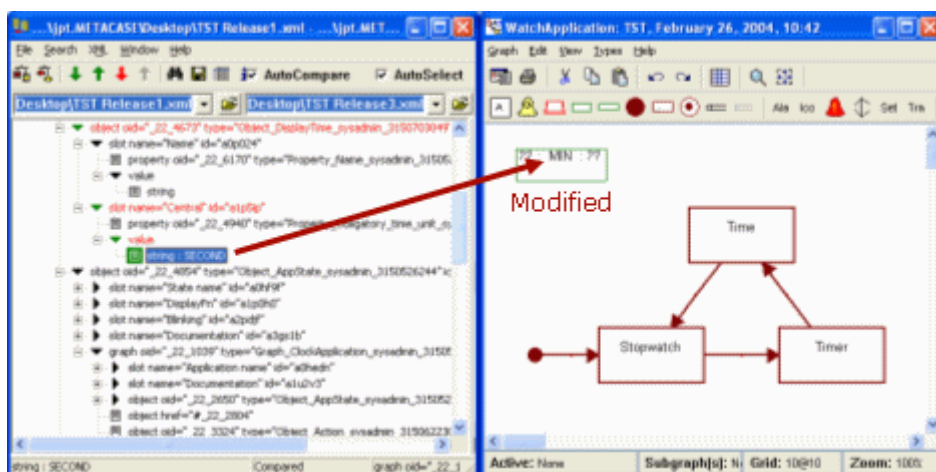
MetaEdit+の API は、アプリケーションの統合に広くサポートされているオープンな規格である SOAP/Web Service/.NET 標準を使用しています。その為、MetaEdit+の機能は、殆ど全てのプログラム言語（Java、C、C++、C#、VisualBasic、Perl 等）プラットフォーム（Windows98 / NT / 2 0 0 0 / XP、Linux、HP-UX、Solaris 等）からアクセス可能です。

## コード実行中の MetaEdit+ 内のモデルのアニメーション表示



エミュレータやプログラミング環境から、API を使って実行されているモデルのトレースやアニメーション表示が可能です。コードが実行されると、対応した MetaEdit+のモデルエレメントが赤いアウトラインでハイライトされます。

## 差分解析：モデルの比較



MetaEdit+の API を使って、モデルとモデルが持っているデータの全ての情報を、XML 形式で出力できます。XML の DIFF ツールを使えば、モデルのバージョンを簡単に比較できます。API コマンドを使えば、変更や追加のあったエレメントをハイライト表示することも可能です。

上の例は 3 種類の変更を示しています。最初は、値を “ seconds ” から “ minutes ” に変更することでプロパティを変更しています。次に、新しい “ Start ” オブジェクトがそのリレーションシップと共に追加されています。最後に、古い “ Start ” オブジェクトがそのリレーションシップと共に削除されています。

## マルチユーザーのモデル作製環境

MetaEdit+は、シングルユーザー、マルチユーザーの両方の環境で実行できます。どちらの場合でも、MetaEdit+の心臓部はオブジェクトレポジトリです。オブジェクトレポジトリには、MetaEdit+の、言語の定義、図表、マトリクス、オブジェクト、プロパティ、選択されたフォントの種類までも含めた全ての情報が保存されています。それゆえ、1つのツールやMetaEdit+の1クライアントがシステム的设计を変更すれば、それを表示している全ての場所で自動的に変更が反映され、整合性が保障され、情報がアップデートされます。同様に、レポート出力やコード生成もレポジトリに保存された情報を直接基づいています。

他のモデル作成環境とは違い、MetaEdit+のマルチユーザー環境は、異なったプラットフォーム上で、異なったライセンス形態を使い、異なったアクセス権限の基で使用できます。

オブジェクトレポジトリは数十億の設計オブジェクトを保存可能です。マルチユーザーの並列処理も非常に快適に行います。ロックされる設計情報を最小にする（設計データがコンフリクトしない範囲でロックアウトを最小にする）ことで、複数のユーザーが同じモデル（同じオブジェクトでさえ）を使って作業できます。

各MetaEdit+のオブジェクトレポジトリには、異なったDSM言語、モデル及びコードジェネレータを保存できる点がユニークです。モデリング言語とコードジェネレータへの最新の修正は、利用可能にするためにコンパイルする必要が無いことから、全てのユーザーがリアルタイムに利用できます。この方法を取ることで、モデルと完全なモデリング言語でさえ、効果的に共有できます。

## サポートされるプラットフォーム

MetaEdit+はプラットフォーム依存しないように設計されており、全ての主要なプラットフォーム上で動作する様にオープンシステムの原則に従っています。

プラットフォーム	ハードウェア	オペレーティングシステム
Windows	Pentium	Windows XP, 2000, ME, 98*
Linux	Pentium	例) RedHat 6.0 以降
Mac OS X	PowerPC Mac (インテル Mac 未対応)	Mac OS X 10.2.x 以降
Solaris	Sun SPARC workstation	Solaris 2.5 以降
HP-UX	HP PA-RISC	HP-UX 10.02 以降

プログラムのインストールにはハードディスクに30MBの空き容量が必要です

MetaEdit+はWindows95とNT4でも動作しますが、マイクロソフトはこれらのOSのサポートを打ち切っています。

## ツールの比較

DSM を使った開発の自動化には、現行のツールとその機能、関連費用及びサポートサービスの利用可否を評価を必要とします。以下の表は MetaEdit+ と他のツールとの比較に役立てるように作成しました。

比較表を完成させるために自由にコピーしてください。書かれている特徴に間違いがあればご指摘ください。

ツールの特徴	MetaEdit+	他のツール
<b>言語の開発</b>		
幾つかの統合されたモデリング言語	✓	
複数のモデル作成者の同時使用	✓	
言語定義ツール (メタモデリング)	グラフィカルとフォームベース	
モデリングとメタモデリングの統合 (言語の定義中に同時にテストが可能)	✓ 統合されたツール	
モデル作成者への言語の自動的な展開	✓	
メタモデルによる言語のルールと制約のカバー	✓ ルールと制約の定義ツール及び、ジェネレータベースの制約と正規表現の記述	
言語定義時のメタモデルのチェック	✓	
メタモデルの部品間のトレーサビリティ	✓ 双方向: “誰が部品を使っているか” と “どの部品が使われているか” のトレース	
メタモデルの拡張・変更	✓ 展開された言語は安全に変更可能	
メタモデルを単一言語又は複数言語に特定する	✓ 両方可能	
メタモデル変更のモデルに対する自動アップデート	✓	
旧言語バージョンのサポート	✓	
多様なメタモデルを扱うためのリポジトリ	✓ オブジェクトのリポジトリ	
メタモデルのインポートとエクスポート	✓ XML とバイナリーフォーマットが利用可能。シンボル、制約、ジェネレータの定義もカバー。	
メタモデルのライブラリ	✓ 70 以上の言語を利用可能	
ステレオタイプ、タグ付き値及び制約を含む拡張 UML	✓ 利用可能。不要な部品を削除したり、新たなタイプを追加することも可能。	
<b>シンボルや表示の定義</b>		
言語表示の規定	✓ シンボルエディターによる描画やシンボルのインポートが可能。	
豊かな表示方法 (ノードに対するアイコンや接続に対するライン以上のもの)	✓ オブジェクト、リレーションシップ、ロール及び、プロパティに対するシンボルを定義可能。	
モデルデータをベースにしたシンボルの変更	✓ ジェネレータベースのシンボル部品、条件付シンボル、正規表現、接続先を指定できるシンボル内のポート等	
シンボルライブラリの利用	✓ 数百の表示部品が利用可能	
シンボルのインポート (エクスポート)	✓ SVG とビットマップをサポート	
プログラム言語やプラットフォームからの表示へのアクセスと修正	✓ SOAP / ウェブサービス API をサポート	

モデリングツール開発の自動化	✓ アイコン、メニュー、ツールバー、印刷、オートレイアウト、ダイアログ、カスタムレイアウト等の機能と共にエディターが利用可能	
<b>ジェネレーター定義</b>		
ジェネレーターとメタモデルの統合	✓	
直接的なモデル - コード変換のサポート	✓ 中間フォーマットは不要	
テンプレート、ビジターパターン、クローラ、複数ストリームのサポート	✓	
構文の強調表示と出力のフィルター処理	✓ ジェネレータエディタ	
ジェネレータのデバッグ	✓ ジェネレータデバッガ	
生成されたコードと手書きのコードの分離	✓	
生成結果内へのプロテクトブロックの定義	✓	
モデルとファイルの複数マッピング	✓	
<b>独自言語を使ったモデル化</b>		
複数種類エディターのサポート	✓ ダイアグラム、マトリクス、テーブルエディター	
モデリング操作のサポート(コピー/ペースト、再利用、置換、グループ化、レイアウト、グリッド、ズーム、階層化等)	✓	
複数のモデル作成者による共同利用のサポート	✓ マルチユーザーとシングルユーザー機能	
メタモデルをベースにしたモデルの自動検証	✓	
モデル作成時と要求時の制約のチェック	✓ メタモデルとモデルチェックレポートをベースに自動チェックが可能	
モデル間でのリンクと再利用	✓ メタモデル内に定義することで可能	
言語バージョン更新時のモデルの自動アップデート	✓ 必要に応じて可能	
古い言語バージョンでのモデルの保存	✓ 必要に応じて可能	
モデル参照ツール	✓ 多様なブラウザ	
ドキュメント生成	✓ RTF/Word と HTML フォーマットの定義済みでカスタマイズ可能なジェネレータ	
メトリクスとモデルのチェック	✓	
モデル部品間でのトレーサビリティ	✓	
モデルのインポートとエクスポート	✓ XML とバイナリーフォーマット。表示も含む。	
モデル間の変換	✓ XML を中間フォーマットとして、或いは API を使って直接、ジェネレータエディタで変換可能。	
プログラム言語 / プラットフォームからのモデルへのアクセスと変更	✓ SOAP / ウェブサービス API を使って、殆どのプログラム言語とプラットフォーム	

	からアクセス可能。	
リバースエンジニアリング	✓ XML インポート又は、API を使って可能	
生成されたコードからモデルへのトレースバック	✓ Live-code 機能：生成されたコードをクリックすることでモデルを表示	
(IDE 上、エミュレータ上、実ターゲット上での)プログラム実行時にモデルをアニメーション表示。	✓ API を使ったアニメーション表示	
UML サポート	✓ UML のメタモデルを付属	
<b>その他</b>		
他のツールとの統合 (IDE、バージョン管理、エミュレータ)	✓ 最も一般的な統合手法 (SOAP と XML) をベースに簡単に統合可能。	
利用と学習が簡単	✓	
他の技術の必要性		
複数プラットフォーム上での実行	✓ 複数プラットフォームのサポート。異種プラットフォーム上でのセットアップもサポート。	
マニュアル	✓ モデリング、メタモデリング、システム管理者向けのマニュアルが付属	

サポートサービス	MetaEdit+	他のツール
メンテナンス	✓	
サポート窓口、ヘルプデスク	✓	
トレーニングサービス	✓ <ul style="list-style-type: none"> <li>- エキスパート向けの DSM 作成トレーニング</li> <li>- モデル作成者向けの DSM 利用トレーニング</li> <li>- Q&amp;A、プロセス、管理のための組織変更トレーニング</li> </ul>	
コンサルティングサービス	✓	



富士設備工業株式会社 電子機器事業部

<http://www.fuji-setsu.co.jp>

〒591-8025 大阪府堺市北区長曾根町1928-1 Tel: 072-252-2128