

# 平成15年度未踏ソフトウェア創造事業

## WideStudio for T-Engine

### 成果概要

開発代表者：平林 俊一

担当 PM：坂村 健

プロジェクト管理組織：社団法人トロン協会

#### 1. 背景

最近では、各社から多様な統合開発環境が発売されるようになり、開発環境は、今や身近なものとなっている。ただ、残念ながら、BTRON や T-Engine をとりまく環境において、Java 環境以外で、どのプラットフォームでも統一された操作性、開発方式で開発できるものは無いのが実情であった。

#### 2. 目的

本プロジェクトではこの点に着目し BTRON・T-Engine プラットフォームにおけるソフトウェア開発効率を改善する開発環境の開発を行った。

フォームでも統一された操作性、開発方式で開発でき、しかも互換性のあるアプリケーションを作成できる点が従来技術との最大の異なる特徴である。WideStudio for T-Engine の特徴の詳細や WideStudio for T-Engine における実際のアプリケーションの構築手順については、次の章において記述する。

#### 3. 開発の内容

平成15年度 IPA 未踏ソフトウェア創造事業 WideStudio for T-Engine プロジェクトでは、マルチプラットフォームで、それもオープンソースの統合開発である WideStudio を BTRON・T-Engine に対応させ、他のプラットフォームのアプリケーションと同様に、BTRON や T-Engine 用のアプリケーションを構築できるようにすることを目的として遂行された。

#### 5. WideStudio for T-Engine の詳細

WideStudio は、FreeBSD、Linux、Solaris、Windows95/98、WindowsNT/2000 上で動作する、デスクトップアプリケーションを構築するための統合開発環境(IDE)である。WideStudio はオープンソースベースのソフトウェアであり、誰もが利用できる。それを T-Engine 向けに適用した。

#### 4. 従来技術との相違

単に BTRON や T-Engine 対応の開発環境を構築するだけではなく、他のプラットフ

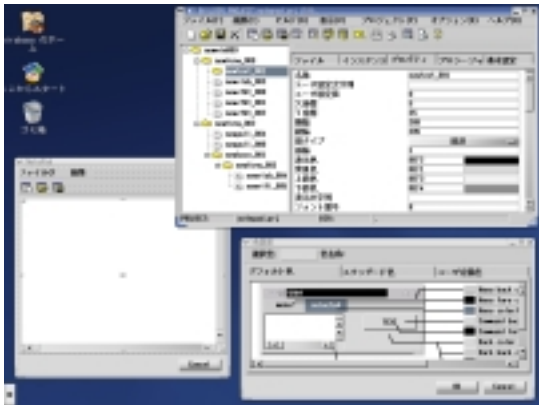


図1 統合開発環境 WideStudio

WideStudio の特長は次の示す通りである。

- ・マルチプラットフォーム
- ・マルチランゲージ
- ・直感的なアプリケーションの構築

### 5.1 マルチプラットフォーム

WideStudio で作成したアプリケーションには、ソースコード互換性があり、例えば、Windows 上で作成した GUI アプリケーションが、WideStudio for T-Engine によりリコンパイルすれば、T-Engine 上で動作する(図2、図3)。また、異なるプラットフォーム間においても、同じ画面ビューや操作性を実現しており、作ったデスクトップアプリケーションは、Windows 上でも、T-Engine 上でも全く同じように実行させることができる。

### 5.2 マルチランゲージ

WideStudio では、C/C++ 以外に Perl/Ruby/Python を用いてデスクトップアプリケーションを構築することができる。作製するアプリケーションに合ったプログラミング言語使用でき、既存のプログラム資産や、プログラミングスキルを持った人

的資産を有効に利用することができる。



図2 Linux 上の WideStudio アプリケーション



図3 T-Engine 上の WideStudio アプリケーション

### 5.3 WideStudio for T-Engine

WideStudio for T-Engine は、BTRON や T-Engine におけるデスクトップアプリケーションを製作するための WideStudio である。WideStudio for T-Engine を使うと

次のような特徴がある T-Engine アプリケーションを製作することができる。

- ・ ネイティブバイナリの高速なアプリケーションを製作可能。
- ・ Java チップのような特別な実行チップ、または JavaVM 等を必要としない。
- ・ 他のプラットフォームとプログラム資産が共有可能。

これらの特徴は、少ないハードウェアリソースで動作しなければならない組み込み系の世界では大きなアドバンテージとなる。

#### 5.4 WideStudio for T-Engine の入手と環境の構築

WideStudio for T-Engine では、Linux を開発環境とし、T-Engine アプリケーションを Linux 上で構築する(将来的には WIN32 プラットフォームにおいても構築可能とする予定である)。ビルドしたアプリケーションを T-Engine に転送し、実行しデバッグする開発形態をとる。

#### 5.5 WideStudio for T-Engine 必須ソフトウェア

WideStudio for T-Engine を動作させるために必要なソフトウェアを、下記に示す。

- ・ gcc/g++ などの Linux 用のコンパイラ
- ・ ソースコードを編集するためのエディタ
- ・ T-Engine 開発環境
- ・ T-Engine T-Shell 開発環境

コンパイラは、Linux に付いて来る Linux 用の gcc/g++ コンパイラと、T-Engine 開発環境用のコンパイラが必要である。また、T-Shell 開発環境も必要と

なるので、注意が必要である。

#### 5.6 WideStudio for T-Engine の入手

WideStudio for T-Engine 下記の公開ページから、WideStudio のソースコードを入手することができる。

<http://www.widestudio.org>

WideStudio のソースコードは、WideStudio のバージョン番号を X.XX-X とすると含んだ ws-vX.XX.X-src.tar.gz というファイル名で公開されている。ソースコードを入手したら、T-Engine 開発環境がインストールされている Linux 機で次のように展開する。

```
cd /usr/local/te
gzip -cd ws-vX.XX.X-src.tar.gz | tar -xvf -
```

#### 5.7 WideStudio for T-Engine のビルド

WideStudio for T-Engine のビルドは 2 つのフェーズからなる。1 つは WideStudio 本体のビルドと、1 つは T-Engine 用のライブラリのビルドである。マニュアルに従って各ビルドを行い、環境を構築する。

- ・ WideStudio 本体のビルド
- ・ WideStudio for T-Engine のビルド

#### 6 . T-Engine アプリケーションの作成手順

WideStudio for T-Engine では、他のプラットフォームでの作成方法と同じように T-Engine アプリケーションを作成することができる。まず、wsbuilder コマンドでアプリケーションビルダーを起動して作業

を行う(図4)

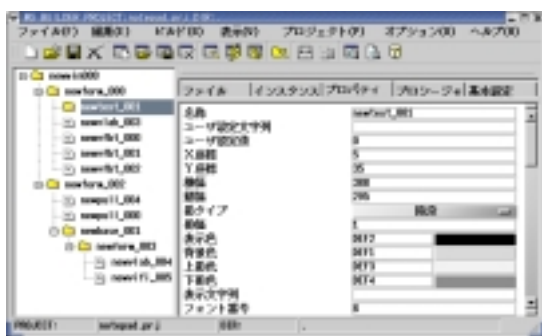


図4 WideStudio アプリケーションビルダ

WideStudio アプリケーションビルダは、下記に示すような特徴を持っている。

- ・プラットフォームに依存しない統一された操作性
- ・拡張ロー風のアプリケーション構築環境
- ・インタラクティブなウィンドウアップレーションの編集
- ・イベント駆動方式によるソースコード記述
- ・メイクファイルの自動生成とビルド機能

### 6.1. プロジェクトとは?

通常、デスクトップアプリケーションには、多くのウィンドウが存在したり、多くのイベントプロシージャが存在する。

WideStudio のプロジェクトは、これらを一元管理し、コンパイルのための環境の保持やメイクファイルの作成、ソースファイルの生成、アプリケーション構築等を行う。

WideStudio でのデスクトップアプリケーションの開発手順は次の様な手順を踏む。

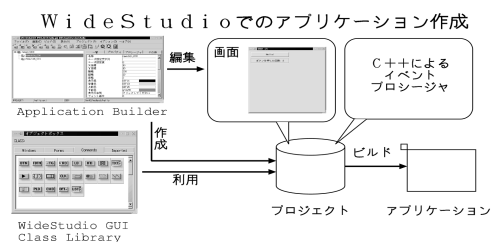


図5 アプリケーション作成の流れ

手順1: 開発する対象となるアプリケーションに対し、プロジェクトを作成する。

手順2: ウィンドウを作成し、部品を配置して編集する。

手順3: イベントプロシージャ(後述)を記述し、部品に張り付ける。

手順4: 最後にビルドし、実行ロードモジュールを作成する。

### 6.2. hello プロジェクトの作成

簡単なアプリケーション hello を例に、WideStudio for T-Engine で実際に T-Engine アプリケーションを作る一連の手順を示す。



図6 hello の画面

まず、wsbuilder を起動し、アプリケーションビルダーの [プロジェクト] メニューの新規プロジェクトを選択し、新規にプロジェクトを作成する。プロジェクト名に hello を指定して、「通常のアプリケーション」を選択する。作成されるプロジェクトファイル名は、hello.prj となる。

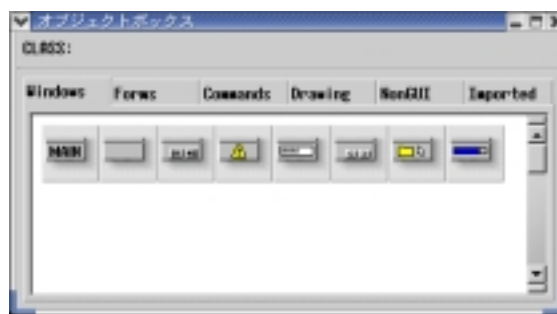


図7 オブジェクトボックス

### 6.3 ウィンドウの作成

プロジェクト作成の次は、hello のベースとなるウィンドウの作成を行なう。ビルダーの [ファイル] メニューの新規ウィンドウを選択し、タイプに「通常のウィンドウ」を指定する。newwin000 の名称を指定し、プロパティを下記のように設定する。

プロパティ	値
オブジェクト名称	newwin000
X 座標	0
Y 座標	0
横幅	240
縦幅	300
タイトル属性	タイトルなし

Commands タブを選択して、ボタンオブジェクトを含むコマンドオブジェクトのグループを表示させる。BTN と書いたアイコン (WSCvbtn/プッシュボタンクラス) をウィンドウにドラッグアンドドロップして配置し、配置されたボタンオブジェクトを選択し、プロパティを次の様に設定する。

プロパティ	値
オブジェクト名称	newvbtn_000
X 座標	10
Y 座標	10
横幅	100
縦幅	30
表示文字列	TEST

次に、作成されたばかりのウィンドウ上に、ボタンオブジェクトを配置する。[表示] メニューの [オブジェクトボックス] を選択すると、次の図7のようなオブジェクトボックスが表示されるので、

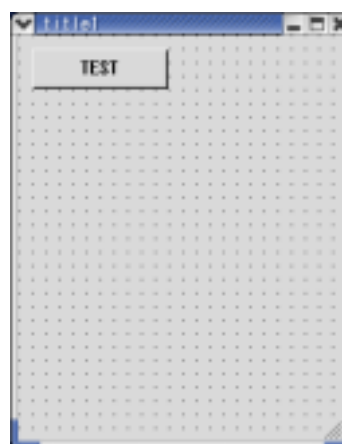


図8 ボタンの形状を整えたところ

## 6.4 イベントプロシージャの作成

イベントプロシージャは、C/C++ 言語で記述し、オブジェクトのアクセスや任意のデータ処理等を行う。そして、作成したプロシージャを配置した部品に対して、起動されるイベントを指定する。例えば、イベントプロシージャを、ボタン部品に対してマウスが押される等のイベントで設定しておく、マウスでボタンが押した時に実行される。ここではボタンを押すことにより、「Hello!」と表示するイベントプロシージャを作成する例を示す。

アプリケーションビルダーで、newbtn\_000 を選択して、[編集]メニューの[プロシージャ編集]の[プロシージャ新規作成]を選択し、次のように設定する。

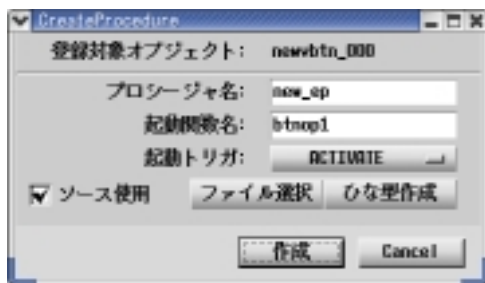


図9 イベントプロシージャの作成

プロシージャ名: new\_ep  
起動関数名: btnop1  
起動トリガ: ACTIVATE

プロシージャ名は、イベントプロシージャを識別するためのもので、自由に付け管理することができる。起動関数名は、起動される C/C++ の関数名を指定する。起動トリガには プッシュボタンを押して放した時に発生する ACTIVATE を指定する。

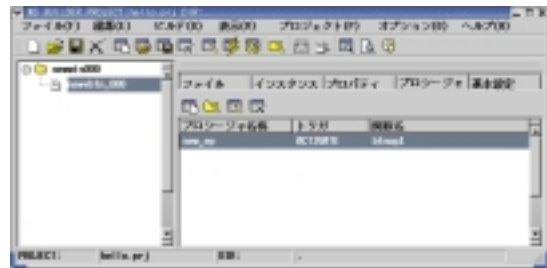


図10 イベントプロシージャの様子

ファイル名 btnop1.cpp で 空の関数 btnop1() が生成されているので、そこに Hello! と表示するように記述 //(A) の部分を追加する。記述 (A) は表示文字列プロパティ WSNLabelString を、「Hello!」に設定する。

```
#include <WSCom.h>
#include <WSCfunctionList.h>
#include <WSCbase.h>
//-----
//Function for the event procedure
//-----
void btnop1(WSCbase* object){
    static long cnt = 0;
    object->setProperty(WSNLabelString,"Hello!"); //(A)
}
static WSCfunctionRegister op("btnop1",(void*)btnop1);
```

## 6.5 hello プロジェクトのビルド

作成した hello プロジェクトをアプリケーションビルダーの [プロジェクト]メニューの [プロジェクト保存] を選択して保存後、アプリケーションビルダーの [ビルド]メニューのビルドオールによりビルドを行う。



図11 ビルド画面

ビルドが成功すると、BTRON / T-Engine 用プロジェクトでは hello.bz が生成される。この hello.bz を T-Engine を T-Engine 開発環境に標準で添付される gterm を用いて次のようにシリアル転送し、次のように設定する。

```
[/SYS] recv -d ../hello/hello.bz  
[/SYS] expf -v hello.bz  
[/SYS] vup -t hello /SYS  
[/SYS] rm hello hello.bz
```

hello.bz のファイルを古パスで指定。  
hello.bz を展開。  
hello を小物に登録。  
なくなった hello を削除。



図 1 2 hello を T-Engine に転送し展開したときの gterm の様子

これで、「サンプルアプリケーション」という題目で小物に hello が登録される。図 1 3 は hello を実行した時の様子を示す。



図 1 3 hello を実行したときの様子

登録時のタイトル文字列「サンプルアプリケーション」は、ビルド時に作成される hello.f を編集し、ビルドすることで変更することができる。

## 7. WideStudio for T-Engine による効果

WideStudio for T-Engine によるアプリケーション構築における効率を次の条件で評価した。

条件：

- (1) タッチパネルによりボタンを押下し、Hello! と表示する。
- (2) 必要なソースコード、Makefile 等は、サンプルのものを用いず、生成されないものは全て記述するものとする。

開発ケース	必要開発時間
ツール無、未経験者	1~2週間
ツール無、経験者	2日~1週間
WideStudio、未経験者	1日
WideStudio、経験者	10分

効率的に BTRON/T-Engine 上で動作するアプリケーションが構築できることを示している。

## 8. 普及の見通し

現在の WideStudio の普及率に関する諸データを下記に示す。

総ステップ数	300Kstep
ページビュー数	55万件
ダウンロード数	累計70万件
推定ユーザ数	5～6万人(国内)、1万人(国外)
開発年数	5年
メーリングリストメンバー数	1100人

韓国語、中国語対応を機に、国外ユーザを10万人に増やす。

国内普及率向上のため、2004年夏までに本プロジェクトに関して書籍を発刊する予定である。

予定書籍名(仮題):

「WideStudio 標準ガイドブック」

WideStudio for T-Engine のてこ入れにより、国内ユーザ数10万人を目指す。

## 9. 参考 URL

### 1. WideStudio 公開ページ

<http://www.widestudio.org>

### 2. WideStudio メーリングリスト Web ページ

<http://www.egroups.co.jp/group/widestudio/>

### 3. SOURCE FORGE WideStudio 公開ページ

<http://sourceforge.net/projects/widestudio/>