

# 組込みソフトウェア統合開発ツール 「SystemDirector Developer's Studio Embedded」

小崎 光義・小泉 健  
大橋 充幸・佐野 建樹

## 要 旨

組込みソフトウェア開発の効率化を行うために、NECではオープンソースソフトウェアのEclipseをベースとする統合開発ツールを開発しました。本製品は、設計、実装、検証というソフトウェア開発の各工程を一貫してサポートしており、一連の作業を途切れなく実行しプロジェクト全体の効率的な開発と品質の向上を実現します。本稿では、組込みソフト統合開発ツールSystemDirector Developer's Studio Embeddedの概要を紹介します。

## キーワード

●統合開発ツール ●UML ●JTAG ●静的検証 ●クロス開発 ●Eclipse

## 1. はじめに

携帯電話、車載機器やデジタル家電などに搭載される組込みソフトウェアの開発は、開発規模の肥大化、製品サイクルの短縮に伴い、ソフトウェアの品質確保や開発効率の向上が大きな課題となっています。その対策として開発ツールを利用することが考えられますが、開発工程の1つ1つを効率化するのではなく、組込みソフトウェア開発のライフサイクル全般の効率化を実現する統合的な開発ツールが求められています。

一方、Enterprise系ソフトウェア開発では、開発言語としてJavaの利用が拡大し、OSS（オープンソースソフトウェア）の統合開発ツールであるEclipseが普及しています。Eclipseは拡張性の高さと無償で利用できるといった特長から、Enterprise系の開発ではデファクトスタンダードになりつつある統合開発ツール基盤です。しかしながら、組込みソフトウェアの開発言語はC言語（一部C++）が中心であり、これまでEclipseを十分活用できる状況にありませんでした。

SDDS/E（SystemDirector Developer's Studio Embedded）は、EclipseをベースとしたC/C++言語対応の組込みソフトウェア開発向けの統合開発ツールです。Eclipseによる統一されたユーザインターフェースで、組込みソフトウェア開発の上流から下流工程までを一貫してサポートすることにより、組込み

ソフトウェアの開発効率と品質を向上させることができます。

## 2. SDDS/Eの概要

SDDS/Eは基盤となるEclipseと、プラグインとよばれる機能拡張モジュール群から構成されます（図1）。中核となるプラグインはOSSのCDTです。CDTはC/C++言語用の実装用プラグインですが、これだけでは組込みソフトウェア開発は困難です。そこで、1) クロス開発を可能とする検証用プラグイン、2) 性能/挙動解析用プラグイン、3) ソースコード検証

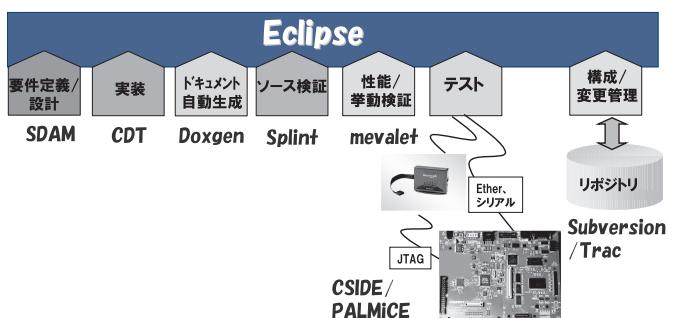


図1 SDDS/E概要

プラグイン、4) ドキュメント生成プラグイン、5) UMLを用いた要件定義・設計用プラグイン、6) 全工程を支援する構成管理ツールとの連携用のプラグインなどを追加しています。Eclipseにこれらのプラグインを追加して統合開発ツールとすることには次の2つの利点があります。

### (1) 開発者ビュー/操作性の統一による習得の効率化

従来は各開発工程でばらばらのツールを利用しなくてはなりませんでしたが、各機能をEclipse上に統合することで開発者ビューや操作性が統一されます。これにより、ばらばらのツールの操作の習得に比べて、効率的な習得が可能です。また、すでにEclipseによる開発経験がある場合には、さらに短期間に操作を習得することが可能となります。

**(2) 開発プロセス/チーム開発における成果物管理を効率化**  
統合開発ツールを利用して開発作業における上流から下流までの開発プロセスの一連の作業を切れ目なく実行可能になります。これにより、設計情報の下流工程への円滑な移行を行うことが可能となります。また、構成管理ツールとの連携することにより個人の作業だけではなく、複数人によるチーム開発での成果物の受け渡しが可能となります。さらには本製品で対応している構成管理ツールでは地理的に離れた拠点との連携も可能であるため、大規模な分散開発を効率化させることができます。

## 3. SDDS/Eの機能

次にSDDS/Eの主要な機能について説明します。

### 3.1 クロス開発機能

組込みソフトウェア用の統合開発ツールと他の統合開発ツールとの最も大きな違いはクロス開発機能です。たとえばWindows上の統合開発ツールの場合、1つのPC上でコンパイル・実行・デバッグを行うことが可能です。それに対して、組込みソフトの場合、ターゲット機器へPCからプログラムをロードしてデバッグするクロス開発機能が必要になります。このためJTAGエミュレータ<sup>①</sup>などのハードウェアを利用してPCとターゲット機器を接続してデバッグすることが一般的です。

一方、EclipseはJTAGエミュレータなどのハードウェアと連携したクロス開発機能を提供していません。そこでSDDS/E

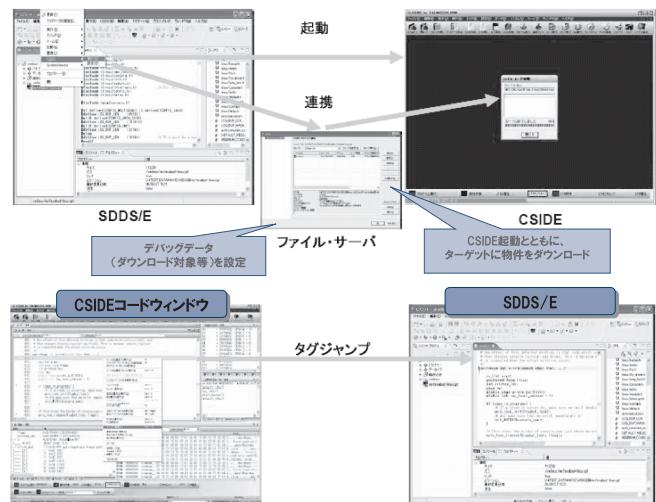


図2 クロス開発機能

では、Eclipseと株式会社コンピューテックスのJTAGエミュレータ「PALMiCE」並びにデバッガ・ソフトウェア「CSIDE」とを連携させて、クロス開発機能を実現しました(図2)。

#### (1) CSIDE、PALMiCEの機能

CSIDE、PLAMiCEはターゲットとしてCPUにARM 7/9/11を搭載した機器に対応しています。プログラムのロード、ソースコードを参照しながらのブレークポイントの設定、ステップ実行などの基本的な機能に加えて性能解析やメモリリーク検出などの機能を提供します。また、停止したポイントから過去にさかのぼって動作を再現させるバケットトレース機能を提供しています。

#### (2) CSIDEとEclipseの統合

PALMiCEはCSIDEから制御され、開発者はCSIDEを操作してデバッグを行います。従来、コーディング・コンパイル作業とCSIDEのデバッグ作業はシームレスに連携できていませんでした。デバッガでエラーを発見すると、何らかのエディタでソースコードを修正して再コンパイルが必要でした。そこでSDDS/Eでは1) EclipseからCSIDE、2) CSIDEからEclipseの双方向の連携機能をサポートしました。1) はEclipseから実行プログラムやデバッグ情報をマウスクリックだけで、CSIDEへ引き渡す機能です。2) はCSIDEで発見した問題の行をEclipseに引き渡してソース修正・コンパイルを可能とする機能です。これらの2つの機

能で、SDDS/EではほとんどEclipseだけで開発している場合と同じようにクロス開発が可能となります。

### 3.2 UMLによる設計効率化支援

現在、組込みソフトウェア開発において最大の課題が「設計品質」であると言われています<sup>2)</sup>。それに対して、多くの組込みソフトウェア統合開発ツールは実装・検証が中心で、設計の機能を標準でサポートしていません。そのため、設計と実装以降の工程で作業の切れ目があり、設計書とソースコードの不一致によるバグの発生や、バグ修正時に設計書の改版が行われないなどの問題があります。

本ツールではUML (Unified Modeling Language) による設計機能を提供しています。UMLとはOMG(Object Management Group)が標準化しているモデリング言語であり、現在もっとも普及しているモデリング言語です。NECではUMLモデリングツールとしてSDAM (SystemDirector Application Modeler) を開発しています。SDAMはEclipseのプラグインとして実装されており、SDDS/EではSDAMの機能（図3）をそのまま利用することでUMLのモデリングを可能としています。以下にSDAMの機能を紹介します。

#### (1) UMLエディタ機能

SDAMはUMLの最新規格であるUML2.0に準拠しています。UML2.0では13種類のモデル図をサポートしていますが、実際の開発ですべての図を使われることはほとんどありません。本製品ではよく使われる11種類の図をサポートしています。

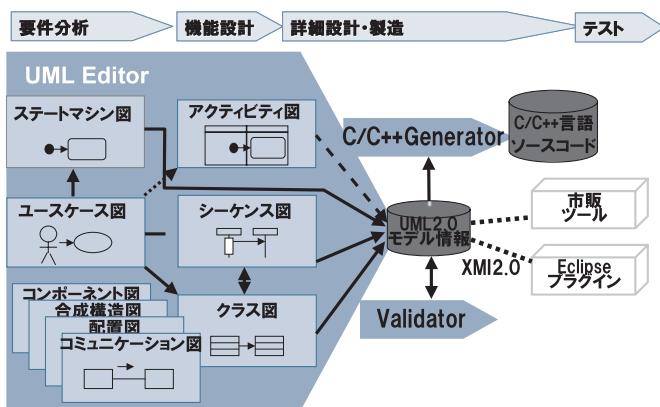


図3 SDAM機能概要

#### (2) UML検証機能

設計の規模が大きくなったり、複雑になったりすると設計バグの増加が考えられます。設計工程でのバグは後工程へ大きく影響を与えますので未然に防ぐ必要があります。本製品ではUML検証機能により二重定義や定義漏れ、モデル図間の不整合などの検証を行い、設計工程でのバグの削減を可能としています。

#### (3) ソースコード自動生成機能

設計書とソースコードの不一致を防ぐためには設計書からのソースコードの自動生成が効果的です。多くのUMLモデリングツールの場合、C++やJavaのソースコード自動生成機能を提供していますが、組込みソフトウェアの場合もっとも多く利用されている言語はC言語です。そのため、本製品で最初に対応した言語はC言語です。

ソースコード生成にはクラス図とステートマシン図を利用します。クラス図からの生成ではソフトウェアの「構造」に関するソースコードを、ステートマシン図からはソフトウェアの「ふるまい」に関するソースコードを生成します。これにより、開発に必要な大部分のソースコードを設計書から生成することができます。さらに、そのソースコードに開発者が追記した場合に、それらのソースコードが保護される仕組みを提供しています。これにより、設計書が修正になりソースコードを再生成した場合に、開発者が追加したソースコードが上書きされないようにとなっています。なお、C言語以外に、C++とJavaのソースコード自動生成機能も提供しています。

### 3.3 ソースコード検証機能

本機能はCDI (コードインスペクション) ツールとEclipseを連携させてソースコードの静的な検証を行う機能です。一般的にソースコードを開発してコンパイルが完了した段階であっても多数のバグが内在しています。これを人手でレビューすることで、テスト工程に進む前に除去することが行われています。CDIツールは、このレビュー工数を削減とともに、人手では検出しにくい問題の検出を行うことが可能です。CDIツールには多数の既存製品があり広く利用されていますが、エディタ・コンパイラと連携されておらず、効率的とはいえませんでした。SDDS/Eでは、CDTのC/C++エディタとCDIツールを連携させることで、効率的に検証を行うこと

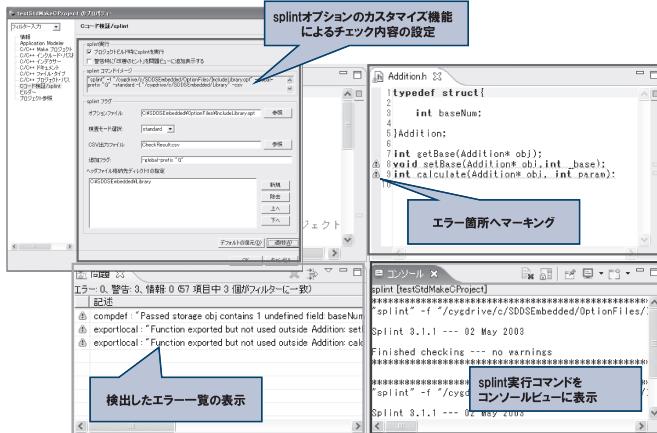


図4 ソースコード検証機能

が可能となっています（図4）。以下に検証機能の特徴を紹介します。

#### (1) コンパイル時に常に検証を起動

開発がほとんど完了の段階でトラブルが発生してからCDIツールを利用している場合がありますが、それでは多数のバグを残したままテストを行うことになります。本検証機能ではソースコードをコンパイルすると同時にCDIツールによる検証を行うことが可能です。これにより開発者はバグの後工程への残留を減少させることができます。

#### (2) 問題一覧とエディタの連動

CDIツールを利用する場合、その結果をみて別に何らかのエディタを起動して行番号を確認しながら修正する必要がありました。また、ソースコード修正中に行番号が変更になった場合には内容を確認しながらの修正が必要でした。本ツールでは問題の一覧のなかからクリックするだけで、CDTの提供するエディタ画面が開いて該当する行にジャンプしてソースコード修正が可能となっています。また、修正中に行番号が変更になった場合にも一覧の行番号が連動するようになっています。これによりCDIツールでの不具合確認からエディタによる修正、コンパイルといったサイクルをシームレスにまわすことが可能となります。

## 4. むすび

本稿では、組込みソフトウェアの品質確保と開発効率の向上を可能とするSDDS/Eについて紹介しました。現在、OSS

のEclipse自体が急激なスピードで機能が進化しています。また、NECにおいても形式検証ツール「VARVEL」を始めとし、開発環境製品はEclipseをベースにしているものが増加しています。SDDS/EではOSSの最新技術を取り入れるとともに、NECで開発している最新機能を提供し、組込みソフトウェアの開発を支援する機能強化を実施していきます。

\*PALMICEおよびCSIDEは株式会社コンピューテックスの登録商標です。

\*Javaは、米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標です。

\*Windowsは、米国およびその他の国における米国Microsoft Corporationの商標または登録商標です。

\*OMG、UML、Unified Modeling Languageは、Object Management Groupの商標または登録商標です。

## 参考文献

- 1) 高田ほか：「リアルタイムOSと組込み技術の基礎」、CQ出版社、2005
- 2) 独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編：「組込みソフトウェア開発における品質向上のすすめ[設計モデリング編]」、アイティメディア社、2006

## 執筆者プロフィール

**小崎 光義**  
システムソフトウェア事業本部  
開発環境技術本部  
マネージャ

**小泉 健**  
システムソフトウェア事業本部  
開発環境技術本部  
主任

**大橋 充幸**  
システムソフトウェア事業本部  
開発環境技術本部

**佐野 建樹**  
システムソフトウェア事業本部  
開発環境技術本部  
シニアエキスパート

●本論文に関する詳細は下記をご覧ください。

## 関連URL

組込みソフト統合開発ツール『SystemDirector Developer's Studio Embedded』：  
<http://www.nec.co.jp/cced/SDDSE/>