

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2010
2010 JSME Conference on Robotics and Mechatronics

ROBOMECH2010 in ASAHIKAWA

ロボティクス・メカトロニクス・フロンティア・ビッグバン
Robotics・Mechatronics・Frontier・Big-Bang

June 13 Sun. - 16 Wed., 2010

Asahikawa TAISETSU Arena

主催 社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



OpenRTM-aist と TECS (TOPPERS Embedded Component System) の連携に関する考察

Consideration about the cooperation of OpenRTM-aist and TECS (TOPPERS Embedded Component System)

李 賢徳 (産総研) ○ 金 湘宰 (デジタルクラフト) 張 敏浩 (デジタルクラフト) 正 安藤 慶昭 (産総研)

Hyundeok LEE, National Institute of Advanced Industrial Science and Technology, hyundeok.i@aist.go.jp

Sangjae KIM, Digital Craft Inc.

Minho JANG, Digital Craft Inc.

Noriaki ANDO, National Institute of Advanced Industrial Science and Technology

To solve the software complexity and maintainability problems, component-based software development is widely adopted in various fields. A component framework for μ ITRON OS named TECS is a kind of static component system. On the other hand, RT-Component framework that has been developed is a kind of dynamic component system. Static component frameworks are suitable for low level software layer that requires real-time capability, safety and reliable quality. In this paper, we propose three architecture to integrate TECS components and RT-Components, and their advantages and disadvantages are discussed.

Key Words: Distributed system, RT コンポーネント, RT ミドルウェア

1. はじめに

現在、実用化を目指し研究が進められているサービスロボットでは、高度なサービスを提供するためにソフトウェアはより複雑化し、同時に、人と共存して動作するためにより一層の信頼性、安全性が求められる。

著者らは、ロボットシステムのコンポーネント指向開発を実現するため、RT ミドルウェアの研究開発を行ってきた。RT ミドルウェアでは、機能要素を RT コンポーネントと呼ばれる、イントロスペクション機能を持つ動的コンポーネントとして作成し、それらの組み合わせで柔軟な RT システムを容易に実現することができる。

一方で、一般的に安全に関わるサブシステム上のソフトウェアでは、リアルタイム性を保証し、動的な構成変更などによる不確定性を排除するため、OS レスまたは規模の小さい組込み OS が用いられる場合が多い。

TOPPERS[1] は機能安全規格 IEC61508 への適合を視野に入れ、形式検証等の取り組みも行われている μ ITRON 系組込み OS である。将来的な機能安全規格への対応を考慮し、RT ミドルウェアを従来の Windows や Linux 等汎用 OS に加え、TOPPERS へ移植し組込み機器での動作を実現した [2]。一方で、TOPPERS には TECS (TOPPERS Embedded Component System) と呼ばれる静的なコンポーネントシステムがあり、組込み機器に搭載するソフトウェアをコンポーネント指向で開発することが可能である。安全に関わる部分では、RT コンポーネントのような動的コンポーネントよりも、コンパイル時に構造が確定される TECS のような静的コンポーネントの方が適している。

著者らは、ロボットの振る舞いを実現する上位系を RT コンポーネント、安全にかかわる下位系を TECS で記述し、両者を連携させることで、柔軟かつ安全なシステムを実現できると考えた。本稿では、TECS と RT コンポーネントを連携する方法について、実現可能ないくつかのアーキテクチャを示す。それぞれについて利点・欠点、実現のしやすさ等の観点から考察を加える。

2. TECS

組込みコンポーネントシステム “TECS (TOPPERS Embedded Component System)” は、TOPPERS プロジェクトのコンポーネント仕様ワーキンググループにおいて仕様の策定が進められている、組込みシステムに適したコンポーネントシステムである。

TECS は、1) 大規模、複雑化する組込みソフトをコンポーネント構造にすることで見通しをよくする (大規模な組込みソフトウェアの見える化)、2) コンポーネントモデルを規定することで、組込み用ソフト部品の仕様定義を容易にし、流通を促進する (ソフトウェア部品の流通性の向上)、3) 組込み向きの分散フレームワークなどフレームワークを実現するためのフレームワークを実現する (分散フレームワークによる分散システムの開発効率化) ことを目指して開発が進められている [3]。

2.1 TECS の構造

TECS は、CDL (Component Definition Language) と呼ばれる記述言語で各種定義を記述、C 言語のコードを自動生成し、コンポーネントをコンパイル時にインスタンス化・接続する、静的なコンポーネントモデルを採用している。TECS で規定しているコンポーネント図を図 1 に示す。コンポーネントの型をセルタイプ、コンポーネントインスタンスをセルと呼ぶ。セル間の接続を実現する関数群の接頭語をシグニチャと呼び、機能を提供する側を受け口、利用する側を呼び口と呼ぶ。

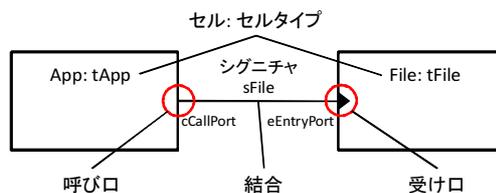


Fig.1 An example of the TECS Component Diagram

2.2 インターフェース定義

下記は、インターフェース定義の CDL による記述である。signature はインターフェースの接頭辞であり、open, close, read, write はシグニチャ sFile のオペレーションである。

```
signature sFile {
  ER open([in,string]const int8_t* FileName,...);
  ER close();
  ER read([out,size_is(ByteToRead)]int8_t* Buffer,...);
  ER write([in,size_is(ByteToWrite)]const void* Buffer,...);
};
```

2.3 コンポーネント定義

下記は、コンポーネント定義の CDL による記述である。tFile が上で定義したインターフェースのサービスを提供する受け口を持つコンポーネントで、tApp が sFile のサービスを利用する呼び口を持つコンポーネントである。

```
celltype tApp{
  call sFile cCallPort;
};
celltype tFile{
  entry sFile eEntryPort;
};
```

2.4 コンポーネントの組み上げ

TECS においてはコンポーネントのインスタンス化とコンポーネント間の接続は静的に行われる。その接続の定義は CDL で以下のように記述される。

```
cell tFile File{
};
cell tApp App{
  cCallPort = File.eEntryPort;
};
```

この記述により、tFile のセルタイプの File コンポーネントがインスタンス化され、一方 tApp セルタイプの App コンポーネントがインスタンス化され、かつ cCallPort の呼び口を、File の eEntryPort へ接続している。

2.5 コード生成

上述したように、CDL で記述されたインターフェース定義とセル間の結合定義に基づき、セルタイプコードのテンプレート、セルのインスタンス化と結合のためのコードが自動生成される。

動的なコンポーネントシステムでは、コンポーネント間の結合は実行時まで未確定であるのに対し、TECS では接続関係がコンパイル時に決定され、インスタンス化や結合にかかる実行時オーバーヘッドをなくすることができる。また、静的に生成されるデータやコードは ROM 領域に置くことができるため、RAM の使用量を削減できるという利点もある。

3. OpenRTM-aist と TECS の連携

RT コンポーネントと TECS を連携させることで、TECS および TOPPERS が提供するリアルタイム性や安全性をロボットシステムに取り入れ、一方で、RT ミドルウェアが提供する共通の枠組みにより、異なる開発言語や OS を使って開発されたモジュールを統合し、柔軟性に富んだロボットを容易に構築できると考えられる。

TECS コンポーネントと RT コンポーネントを連携させる方法には、いくつかの方法が考えられる。以下、3つの案について、実現のしやすさや相互運用性の透過性等の観点から議論を行う。

3.1 TECS 用データポートによる接続 (案 1)

RT コンポーネントのデータポートは、任意の通信インターフェースによるデータ通信方式の拡張が可能な構造となっている。図 2 に示すように、TECS の受け口・呼び口に対してデータの送信、または受信を行うことのできる RT コンポーネントのデータポートを実装することで、RTC と TECS コンポーネント間の通信を実現することができる。

この方法は実現が簡単であるが、TECS の領域と RTC の領域が明確に分離されるため、ツール等の対応が難しいという問題がある。

3.2 doil による接続 (案 2)

多様な通信ミドルウェアを介した通信を実現するフレームワークとして doil (Distributed Object Infrastructure Layer) を開発している。開発中の doil は CORBA (Common Object Request Broker Architecture) と Ice (Internet

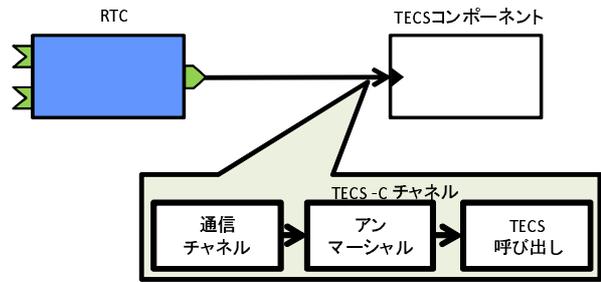


Fig.2 Extended DataPort for TECS RPC communication

Communication Engine) による通信を実現しているが、図 3 に示すように、TECS の RPC を doil の枠組みに取り入れることで、シームレスな通信が実現可能である。しかしながら、実現が難しく、コード量も多くなるため、組込み機器上に搭載するのが難しくなるという問題点も懸念される。

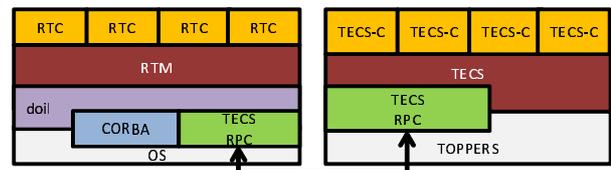


Fig.3 Seamless Integration by using Doil

3.3 CORBA による TECS RPC の実現 (案 3)

RtORB[2] は組込み機器等省メモリ環境のために実装された軽量 CORBA である。RtORB は TOPPERS にも移植されており、TECS で用いる RPC として、この CORBA を利用することができれば、上位の RTC とシームレスに接続することが可能である (図 4)。

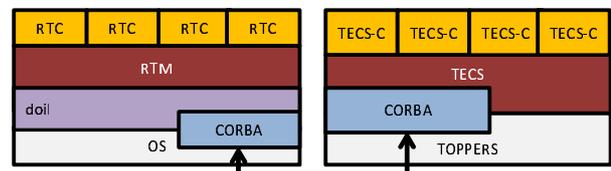


Fig.4 Seamless Integration by using CORBA for TOPPERS

ただし、TECS のコンポーネントモデルと、RTC のコンポーネントモデルが異なるため、同一のツールで両者をシームレスに扱うことは難しいと考えられる。

4. おわりに

本稿では、TECS の安全性、リアルタイム性と OpenRTM-aist の柔軟性、再利用性を併せ持つ新たなロボットプラットフォームの実現のため、RT コンポーネントと TECS の連携方法を 3 つ提案し、それぞれの方法について考察を加えた。

RT コンポーネントと TECS の連携方法として、第 1 段階として案 1 を採用し実装と評価を行うとともに、評価結果に基づき他の案 2 または案 3 についても実装を行う予定である。

文献

- [1] 齊藤他, “コンポーネントソフトウェアに対するハザード分析手法の検討”, 第 5 回システム検証の科学技術シンポジウム予稿集, No.59, pp.65-76, 2008
- [2] 安藤他, “μITRON ベース組込みシステムのための RT ミドルウェア”, 日本機械学会 ロボティクス・メカトロニクス講演会 2009, p.2A1-C14, 2009
- [3] 組込みコンポーネントシステム TECS 仕様書 (NPO 法人 TOPPERS プロジェクトコンポーネント仕様ワーキンググループ), 暫定第一版 (V1.0.0.5)