コンポーネント指向ディペンダブルシステム開発に向けて
– 機能安全の観点からみた RT ミドルウエア –

産業技術総合研究所 安藤慶昭, ○中坊嘉宏, ビグズ・ジェフ, 大場光太郎

Towards component-based dependable system development
– RT-Middleware from the view of functional safety –

Noriaki ANDO, *Yoshihiro NAKABO, Geoffrey BIGGS, Kotaro OHBA, AIST

Abstract—
In dependable systems, which have to be guarantee safety including robotic systems, the software development according to the functional safety standard such as the IEC61508 standard is required. In this paper, the component based software development with RT-Middleware is discussed from a viewpoint of the modular software development based on coding rules which become important from now on.

Key Words: Functional safety, RT-Middleware, modularization

1. はじめに

人と共存する次世代サービスロボットを実現、実用化するうえで、安全性を保証できるディペンダブルなシステムをいかに実現するかが重要な課題となっている。

システムの安全を確保する機能 (安全関連系: Safety related system, SRS) をリスクと許容目標から構成する考え方を機能安全 (Functional safety) と呼び、電気・電子・プログラマブル電子 (E/E/PES: Electric/Electronic/Programable Electronic) の機能安全にに関する国際規格としては IEC61508[1] が知られている。人と直接接触する形で利用される次世代サービスロボットは、誤作動により人間を死傷させる危険性があるため、こうしたロボットを実際に販売する際には、機能安全の考え方に基づきシステムを構成する必要がある。

IEC61508 では、ソフトウエアに対して安全ライフサイクルと V 字モデルによるシステム開発、ソフトウエア作成手法 (ベストプラクティス) を対象システムの安全度水準 (SIL: Safety Integrity Level) に応じて適用することが求められている。これらの手法の多くは一般的ソフトウエア開発に役立つものの、ロボットシステム全般に対しては制約が強すぎたり、認証を得るために膨大な工数とコストがかかるため、ロボットを構成するすべてのソフトウエアに適用するのは現実的ではない。

本稿では、次世代ロボットの機能安全ソフトウエアを効率的に開発する手法として、システムを安全関連系と非安全関連系から構成するアーキテクチャ、および安全関連系のソフトウエアをモジュール化されたソフトウエアの統合により構築する手法について検討する。

2. ソフトウエアの機能安全

システムの安全を確保するための機能安全を実現する部分を安全関連系と呼ぶ。機能安全規格 IEC61508 において、ソフトウエアとはアプリケーションプログラムのみならず、ミドルウエア、OS、ドライバやファームウエア、ツール、コンパイラ、入力すべてを含むと定められている。すなわち、認証を受けようとするソフトウエアは、それを搭載する OS や、ソフトウエアをコンパイルするコンパイラも認証を受けるか、すでに認証を受けたものを使用する必要がある。

機能安全規格 IEC61508 においては、対象システムのリスクアセスメントを行い、危害の重篤度と発生率に基づき、その低減のために要求される安全技術水準 (SIL: Safety Integrity Level) 等の規定が定められている。一般に危害の重篤度が高くより高度な安全技術水準が求められるものは SIL レベルが高く、重篤度が低いものについては SIL レベルが低くなる。

さらにこの SIL のレベルに基づいて、ソフトウエアを構築する際に求められる手法 (ベストプラクティス) が定められている。SIL レベルが高くなるにつれ、より厳密な手法を適用することが求められる。手法の一例としては、形式手法や半形式手法の適用といった安全要求仕様に関する手法や、コーディング規約の使用、動的メモリ確保やオブジェクトの使用の制限といった設計に関する手法、境界値分析に基づくテストといった試験に関するものなど多岐にわたる手法が 100 項目程度定められている。

さらに、ソフトウエアの安全要求仕様から始まり、システム設計、モジュール設計、およびそれらのテストを階層的に行う V モデルに基づく開発ライフサイクルに従って開発されることが要求される。

3. 安全関連系ソフトウエア

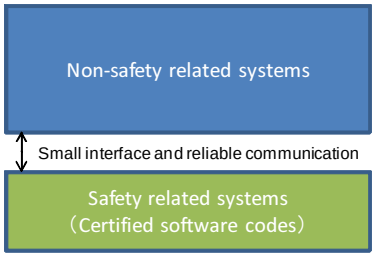

Fig.1 Safety related systems and non safety related systems.

機能安全規格に基づきロボットシステムを構築する

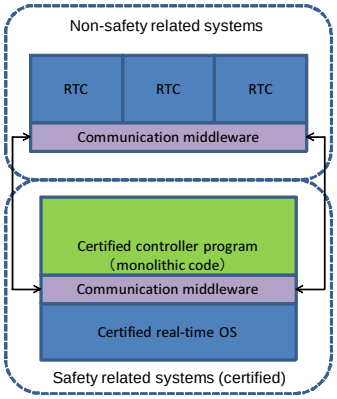

Fig.2 RTC based non-SRS and monolithic SRS.

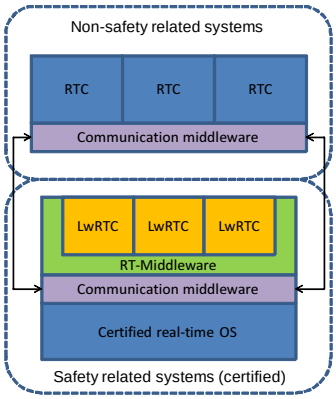

Fig.3 RTC based non-SRS and LwRTC based SRS.

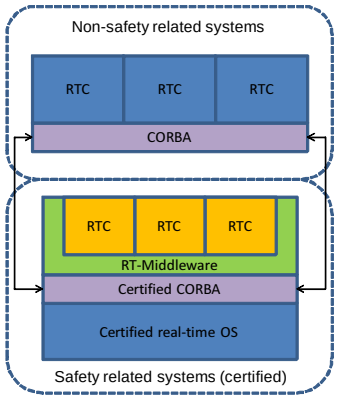

Fig.4 RTC based non-SRS and RTC based SRS.

ことを考えてみる。システムのソフトウエアは、安全関連系と非安全関連系からなり、これらは異なる CPU 上で動作させ、両者を何らかの通信によって接続することで分離することが望ましい (図 1)。

安全関連系と非安全関連系のソフトウエアは、上述したように、それぞれを分離し、かつ安全関連系のソフトウエアは認証可能なプロセスを経て作成される必要があるため、できるだけ単純な構成にする必要がある。一般的には OS を使用しないか、コード量の少ない単純な OS を利用する、あるいは VxWorks や QNX など認証済みの OS を利用する。また、ライブラリやミドルウエア等も極力利用しないか、最低限のものを利用するに留める必要がある。

一方で、ロボットシステムは他の機械システムに比べ、多数のセンサとアクチュエータを持ち、機能安全を実現する際にもより複雑なソフトウエアが求められる可能性がある。複雑化するソフトウエアの構築に対処する方法としては、モジュール化やオブジェクト指向などの手法を利用することが一般的であり、こうした方法は上述のベストプラクティスにおいても推奨される手法である。

著者らはロボットシステムをモジュール化された多数のソフトウエアコンポーネントの集合として構築するためのプラットフォーム、RT ミドルウエアをこれまで提案し、その実装として OpenRTM-aist をオープンソースで公開してきた。OpenRTM-aist は国際標準である OMG RTC (Robotic Technology Component) 仕様に準拠し、分散オブジェクトミドルウエアである CORBA の上に構築されたモジュール化のためのフレームワークであり、動的システム構成の変更が可能であることを特徴とするミドルウエアである。

しかしながら、こうした動的特性は機能安全規格のベストプラクティスによって推奨されない多くの手法が利用されており、これをそのまま安全関連系に適用することは困難である。以上から、ロボットシステムの複雑さに対応するため、安全関連系ソフトウエアをモジュール化技術を利用しつつ、機能安全規格のベストプラクティスに適合する形で実現する方法について考えてみる。

4. モジュール化安全関連系

もっとも単純な方法として、図 2 に示す、非安全関連系を RTC で構築し、安全関連系を従来の方法でモノリシックに構築する方法がある。しかしながら、システムが複雑化するにつれ、安全関連系も複雑になり、改編、改修や拡張に際しては多大な手間とコストが必要となる。

第 2 の方法としては、図 3 に示す、非安全関連系を RTC で構築し、安全関連系には、静的なモジュール化フレームワークとしての軽量 RTC (Lightweight RTC: LwRTC) を利用する方法がある。軽量 RTC は RTC から動的構成機能を取り除き、コンパイル時に接続や設定を静的に決定する。共通インターフェースに基づき機能要素ごとにモジュール化されているため、モジュール間の依存性を少なくでき、システムの変更に際しても認証すべき対象を最小限に抑えられる。安全関連系と非安全関連系間の通信は、静的なプロトコルを利用する単純な通信ミドルウエアを利用する必要がある。

第 3 の方法としては、図 4 に示す、非安全関連系、安全関連系ともに RTC で構築する方法である。安全関連系においても、従来の RTC とほぼ同等の機能を利用できるが、機能安全規格に従えば、CORBA や RT ミドルウエアの動的機能の大部分は利用することはできない。さらに、通信部分を担う CORBA についても機能安全規格に従うため、ベストプラクティスに従った再実装が必要となり、動的な動作を除去することにより、CORBA としての特徴的な機能の大部分が制限されることになる。

5. おわりに

本稿では機能安全規格に従ったロボットシステムをコンポーネント指向で構築する方法について議論した。RT ミドルウエア/RT コンポーネントを利用した機能安全システムとして、3 種類のアーキテクチャを提案した。今後は、これらのアーキテクチャをより詳細に検討し、コンポーネント指向の機能安全ロボットシステムのためのフレームワークを実装し評価する予定である。

参考文献

[1] Functional safety of electrical / electronic / programmable electronic safety-related systems, IEC 61508, 2005