

構成要素の連携動作に着目した産業用ロボットの開発： IEEE 1220 におけるアーキテクチャ設計プロセスおよび UPPAAL によるモデル検査を融合した開発手法の提案

加藤淳^{*}, 浦郷正隆^{*}, 狼嘉彰^{*}

Development of Industrial Robot focusing on Cooperating Behavior among Robot Elements:
A Methodology for combining Architectural Design Process in IEEE 1220 and
Model Checking by UPPAAL

Atsushi Katoh^{*}, Masataka Urago^{*} and Yoshiaki Ohkami^{*}

On developing industrial robots, there are some cases that flaws are incorporated in designing cooperating behavior among elements in an industrial robot, due to complexity of the industrial robot. The cooperating behavior among elements in the industrial robot is generally verified in a system test which is conducted by combining element products on a final development phase. There are impacts to the whole development of the industrial robot when inconsistency of cooperating behavior among the elements is detected in the system test. Therefore, the cooperating behavior among the elements in the industrial robot must be surely developed on an early phase in the development of the industrial robot. This paper proposes a methodology for decomposing a system specification for an industrial robot into element specifications and interface specifications for elements. Consistency of cooperating behavior among their specifications is also verified. The proposed methodology is applied on a system design phase which is an early phase in development of the industrial robot. It is composed of architectural design method in IEEE 1220 and model checking by UPPAAL. It is adopted to a project for developing the industrial robot. As a result of adopting the methodology, it is demonstrated that the methodology is effective for developing the industrial robot from the point of view of QCD.

Key Words : industrial robot, cooperating behavior, architectural design process, model checking

1. はじめに

産業用ロボットとは自動制御によるマニピュレーション機能または移動機能を持ち、各種の作業をプログラムによって実行できる機械である¹⁾。日本は世界に名立たるロボット大国である。世界における産業用ロボットの総稼働台数のうち約 35%が日本で使用されている²⁾。また、日本国内における産業用ロボットの生産額は、2006 年には 7,300 億円に達している³⁾。産業用ロボットは日本におけるモノづくりの代名詞である自動車産業や電子部品産業などの製造業を支えている。

近年、産業用知能ロボット⁴⁾の登場など産業用ロボットの

複雑化が進んでいる。それに伴い産業用ロボットにおける信頼性の確保が課題になっている。産業用ロボットをシステムとして捉えた場合、産業用ロボットはロボットシステムやセンシングシステムなどのシステムが産業用ロボットを構成するサブシステムとして高次に結びついた複雑システムである。複雑システムではシステムの機能を実現するために、システムを構成する要素の処理がインタフェースを介し、他構成要素の処理と密に連携する。本論文ではこれを構成要素による連携動作と定義する。また、本論文ではシステムの仕様に対してシステムの構成要素による連携動作が正しく行われることを連携動作が整合すると定義する。産業用ロボットではロボットサブシステムやセンシングサブシステムなどの構成要素による連携動作が整合していることが重要である。しかし、産業用ロボットの開発においては、その複雑性のために構成要素による連携動作の設計に誤りが生じる場合がある。通常、システムの構成要素による連携動作は、システム開発の終盤に実施される構成要素の実プロダクトを組み合わせたシステム試験で確認される。システム試験において連携動作の不整

^{*}慶應義塾大学大学院

システムデザイン・マネジメント研究科

神奈川県横浜市港北区日吉 4-1-1

(E-mail: katoh.atsushi@z7.keio.jp)

^{*}Keio University Graduate School of System Design and Management

(Received June 28, 2010)

合を検出した場合、システム開発の上流工程に立ち戻り連携動作を再設計する必要がある。その際、開発期間の長期化および改修によるコスト増を招く。また、システム開発の終盤という設計自由度が小さい中での連携動作の設計変更により、システムの信頼性が低下する可能性がある。産業用ロボットの開発においては、市場競争力の観点から開発期間の長期化およびコスト増、信頼性の低下を避けなければならない。

本論文では産業用ロボット開発の上流工程であるシステム設計段階において、産業用ロボットのシステム仕様を連携動作が整合している産業用ロボットにおける構成要素の仕様および構成要素間のインタフェース仕様に分解する開発手法を提案する。われわれは文献5)においてシステムエンジニアリングにおけるアーキテクチャ設計手法および形式手法の1つであるモデル検査の構成方法を示した。本論文では文献5)に示す構成方法に則って実現される産業用ロボットに関する開発手法の実例および産業適用の具体例を示す。産業用ロボット開発の上流工程において、構成要素による連携動作を確実に作り込むことにより産業用ロボットにおける信頼性の向上を図る開発手法は提案されていない。産業用ロボットの構成要素による連携動作を確実に作り込む手順を具体的に示すことで産業界における産業用ロボットの開発に貢献する。提案手法を実現するにあたり、IEEE 1220におけるアーキテクチャ設計プロセス⁶⁾およびモデル検査ツールUPPAAL⁷⁾によるモデル検査⁸⁾を適用する。アーキテクチャ設計とはシステムの構成要素に対しシステムに要求される機能・性能を配分し、構成要素の仕様および構成要素間のインタフェース仕様を明確化する技術である。IEEE 1220に規定されるプロセスに従いアーキテクチャ設計を実施する。それにより、産業用ロボットのシステム仕様を構成要素の仕様および構成要素間のインタフェース仕様に効率的かつ確実に分解することができる。モデル検査とはシステムの状態遷移を表すモデルに対して、与えられた性質が成り立つか否かを計算機により網羅的に検証する技術である。アーキテクチャ設計により導出された構成要素の仕様および構成要素間のインタフェース仕様において、構成要素の連携動作に関する部分に着目しUPPAALによるモデル検査を適用する。モデル検査によって連携動作が満たすべき性質が成り立つかどうかをしらみつぶしに確認する。それにより、複雑な状態で発生する連携動作の不整合を検出することができる。連携動作が整合している仕様を基に各構成要素の開発を行うことにより、産業用ロボットにおける信頼性の向上が期待できる。

本論文の構成はつぎの通りである。2章において、提案手法を実現するIEEE 1220におけるアーキテクチャ設計プロセスおよびUPPAALによるモデル検査を述べる。3章において、提案手法を述べる。4章において、提案手法を適用し不

定形な剛体を運搬する産業用ロボットを開発した事例を具体的に述べる。5章において、開発事例を基に産業用ロボット開発における提案手法の有効性および課題を考察する。6章において、本研究の関連研究を述べる。7章において、本論文をまとめる。

2. 提案手法を実現する技術

提案する開発手法はつぎに示す2つの機能を有する。

- 産業用ロボットのシステム仕様を産業用ロボットにおける構成要素の仕様および構成要素間のインタフェース仕様に分解する機能
- 構成要素の仕様および構成要素間のインタフェース仕様における連携動作が整合していることを検証する機能

機能a)を実現するにあたり、IEEE 1220におけるアーキテクチャ設計プロセスを適用する。また、機能b)を実現するにあたり、UPPAALによるモデル検査を適用する。

つぎに、IEEE 1220におけるアーキテクチャ設計プロセスおよびUPPAALによるモデル検査を述べる。

2.1. IEEE 1220におけるアーキテクチャ設計プロセス

アーキテクチャ設計プロセスは、システムエンジニアリング⁹⁾標準に定められている。システムエンジニアリングとは与えられた費用および期間内で必要な品質を満たすシステムを実現する技術体系である。アーキテクチャ設計を含むシステムエンジニアリングは、主に軍事、航空・宇宙分野で研究が始まり、システム開発のBest Practiceを蓄積・反映しながら標準化されている。Fig. 1にアーキテクチャ設計プロセスを示す。アーキテクチャ設計は機能設計および物理設計で構成される。機能設計のインプットはシステム仕様である。機能設計とはシステム仕様に定義される機能を分割・詳細化し、分割・詳細化した機能に対しシステム仕様に定義される性能を配分する作業である。機能設計のアウトプットは分割・詳細化した機能および性能である。それらは物理設計のインプットとなる。物理設計とはシステムを構成する要素を具体化し、機能設計において分割・詳細化した機能および性能を構成要素に配分する作業である。物理設計のアウトプットはシステムにおける構成要素の仕様および構成要素間のインタフ

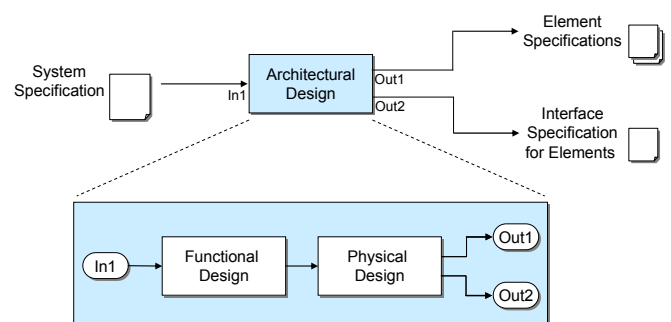


Fig. 1 Process of Architectural Design

エース仕様である。

アーキテクチャ設計に関する詳細なプロセスは、システムエンジニアリング標準である IEEE 1220 に定められている。Fig.2 に IEEE 1220 に規定される機能設計のプロセスを示す。IEEE 1220 における機能設計のプロセスは、IEEE 1220 第 6

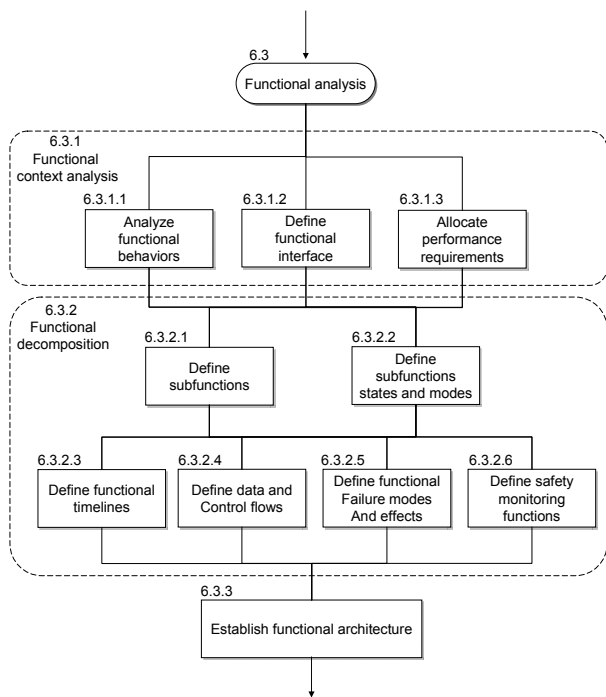


Fig. 2 Process of Functional Design in IEEE 1220⁶⁾

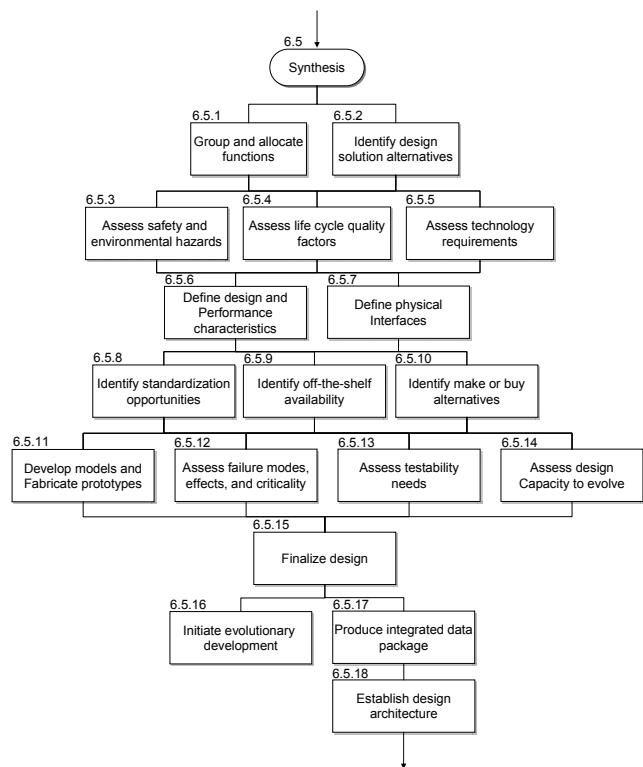


Fig. 3 Process of Physical Design in IEEE 1220⁶⁾

章 3 節 Functional analysis に定められている。Fig.3 に IEEE 1220 に規定される物理設計のプロセスを示す。IEEE 1220 における物理設計のプロセスは、IEEE 1220 第 6 章 5 節 Synthesis に定められている。Fig.2 および Fig.3 において番号が付与されたボックスは、実施すべきタスクを表す。IEEE 1220 にはそれぞれのタスクで実施すべき作業内容が定められている。

IEEE 1220 は、さまざまな産業ドメインにおいて適用され成果を上げている¹⁰⁾。そのため、IEEE 1220 の適用には一定の有効性が保証される。一例を挙げるとアメリカ航空宇宙局 (NASA) では IEEE 1220 の有効性を認め、宇宙機や宇宙機を管制する地上システムなどを開発する際の要求に IEEE 1220 のエッセンスを組み込んでいる¹¹⁾。Fig.2 および Fig.3 の番号に従い IEEE 1220 に定められる作業を実施することで、産業用ロボットのシステム仕様を効率的かつ確実に構成要素の仕様および構成要素間のインタフェース仕様に分解することができる。

2.2. UPPAAL によるモデル検査

モデル検査とは形式手法¹²⁾のひとつに位置付けられる検証技術である。モデル検査は 1980 年代初頭より研究が開始され、近年、ソフトウェア開発において普及が進んでいる。Fig. 4 にモデル検査の概要を示す。モデル検査のプロセスはモデルの作成、検査式の作成、モデル検証の実施、検証結果の分析という 4 つの作業に大別される。まず、適用するモデル検査ツールの表現形式に従って、検査対象となるシステムの仕様などを基に検査対象の状態遷移をモデル化する。つぎに、適用するモデル検査ツールの表現形式に従って、検査対象が満たすべき性質を表す検査式を作成する。そして、計算機上のモデル検査ツールに対して、モデルと検査式を入力しモデル検証を実施する。モデル検証ではモデルが実現しうるすべての状態遷移において、モデルが検査式を満足するか否かを網羅的に検証する。その後、モデル検証の出力結果を基に、モデルと検査式との適合結果を分析する。モデルに対し

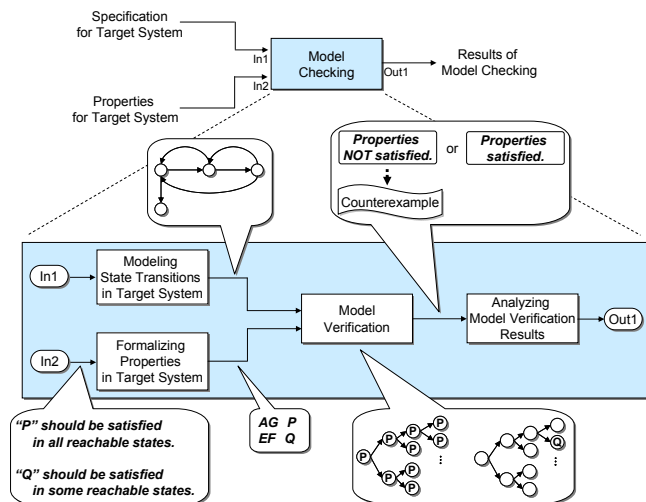


Fig. 4 Overview of Model Checking

て検査式が適合しない場合、検査式が成り立たない条件に至るまでのモデルの状態遷移が反例として出力される。反例を分析しモデルに誤りが無い場合、モデルの基になった検査対象の仕様に誤りがあることを意味する。

モデル検査における計算モデルの代表的な種類として、有限オートマトン¹³⁾ および時間オートマトン¹⁴⁾ が挙げられる。有限オートマトンとは有限個の状態、状態遷移、状態遷移時における動作の組み合わせからなる振る舞いのモデルである。時間オートマトンとは有限オートマトンに時間変数を導入したモデルである。産業用ロボットは決められた時間内に処理を終了する必要があるなどの時間制約を有する場合が多い。したがって、提案手法では時間的な制約を評価することが可能な時間オートマトンに対応したモデル検査ツールであるUPPAALを適用する。

UPPAALにおけるモデルの作成は、GUI (Graphical User Interface) を用いて直感的に行うことができる。Fig. 5にUPPAALモデルの例を示す。Fig. 5において◎は初期状態、○はその他の状態を表す。矢印は状態遷移、矢印に付記される条件式 ("==") は遷移条件を表す。矢印に付記される代入式 ("=") はモデル上の変数に対する値の設定を表す。UPPAALにおける検査式は時相演算子 (G: 常に, F: いずれ), パス限量子 (A: すべての状態遷移のパスで, E: あ

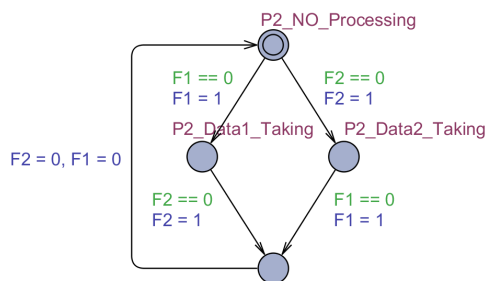


Fig. 5 Sample Model on UPPAAL



Fig. 6 Sample of Counterexample

る状態遷移のパスで), 論理演算子 (OR, AND, NOT) を組み合わせた時相論理¹⁵⁾ で表現される。UPPAALにおける反例の分析にも GUI を用いることができる。Fig. 6にUPPAALにおける反例の出力例を示す。Fig. 6上部に示すウィンドウには、検査式が成り立たない状態に至るまでの状態遷移のパスが出力される。Fig. 6上部のウィンドウにおいて状態遷移のパスを遡る場合、それに対応する状態遷移がFig. 6下部に示す状態遷移モデルに表示される。UPPAALの詳細については文献¹⁶⁾ にまとめられている。

産業用ロボットの構成要素において連携動作の状態遷移にデッドロックなどの不整合が存在する場合、産業用ロボットの安全性に対して致命的な影響を与える可能性がある。UPPAALによるモデル検査を適用して連携動作に関する状態遷移を網羅的に検証する。それにより、連携動作の不整合を検出し連携動作における状態遷移に起因する危険性を排除することができる。

3. 提案手法

本章では、IEEE 1220におけるアーキテクチャ設計プロセスおよびUPPAALによるモデル検査を融合した開発手法を述べる。Fig. 7に提案手法を示す。提案手法のインプットは産業用ロボットのシステム仕様である。提案手法のアウトプットは連携動作が整合している産業用ロボットにおける構成要素の仕様および構成要素間のインタフェース仕様である。Fig. 1で表されるIEEE 1220におけるアーキテクチャ設計は、Fig. 7左側の網掛けボックスに該当する。Fig. 4で表されるUPPAALによるモデル検査は、Fig. 7右側の網掛けボックスに該当する。IEEE 1220におけるアーキテクチャ設計のアウトプットおよびUPPAALによるモデル検査のインプットをシームレスに接続するために、アーキテクチャ設計およびモデル検査の間につき示す作業を設定する。

- 構成要素の連携動作に関する仕様の抽出 (Fig. 7中央上部の白抜きボックス)
- 構成要素の連携動作が満たすべき性質の抽出 (Fig. 7

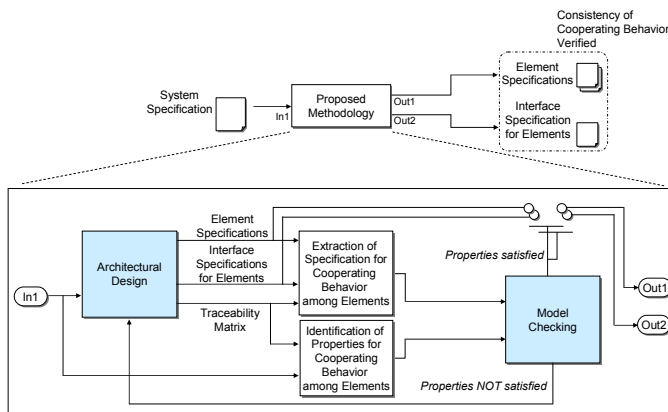


Fig. 7 Process of Proposed Methodology

中央下部の白抜きボックス)

つぎより提案手法のプロセスに従い、その詳細を述べる。

3.1. アーキテクチャ設計

産業用ロボットのシステム仕様をインプットとして、IEEE 1220に定められるプロセスに従いアーキテクチャ設計を実施する。アーキテクチャ設計のアウトプットは、産業用ロボットにおける構成要素の仕様および構成要素間のインタフェース仕様である。また、アーキテクチャ設計の過程において、システム仕様および構成要素の仕様におけるトレーサビリティマトリクスを作成する。Fig. 8に4章で述べる不定形剛体運搬ロボットシステムの開発におけるトレーサビリティマトリクスを例示する。不定形剛体運搬ロボットシステムの仕様がシステム仕様に対応する。ロボット制御サブシステムおよび形状測定サブシステムの仕様が構成要素の仕様に対応する。トレーサビリティマトリクスには仕様が付与した項番単位で、システム仕様とそれをブレイクダウンした構成要素の仕様との対応関係をまとめる。

3.2. 構成要素の連携動作に関する仕様の抽出

アーキテクチャ設計のアウトプットである産業用ロボットにおける構成要素の仕様、構成要素間のインタフェース仕様、システム仕様および構成要素の仕様におけるトレーサビリティマトリクスをインプットとして、連携動作に関する仕様を抽出する。本作業のアウトプットは構成要素の仕様および構成要素間のインタフェース仕様から抽出した連携動作に関する仕様である。Fig. 9に4章で述べる不定形剛体運搬ロボットシステムにおける連携動作に関する仕様を例示する。ロボット制御サブシステムおよび形状測定サブシステムの仕様が構成要素の仕様に対応する。サブシステム間のインタフェース仕様が構成要素間のインタフェース仕様に対応する。構成要素の仕様から連携動作に関する仕様を抽出する際、システム仕様および構成要素の仕様におけるトレーサビリティマトリクスを用いる。トレーサビリティマトリクスを用いて、システムの仕様が複数の構成要素に対応付けられる仕様を構成要素による連携動作の仕様として抽出する。Fig. 8の場合、

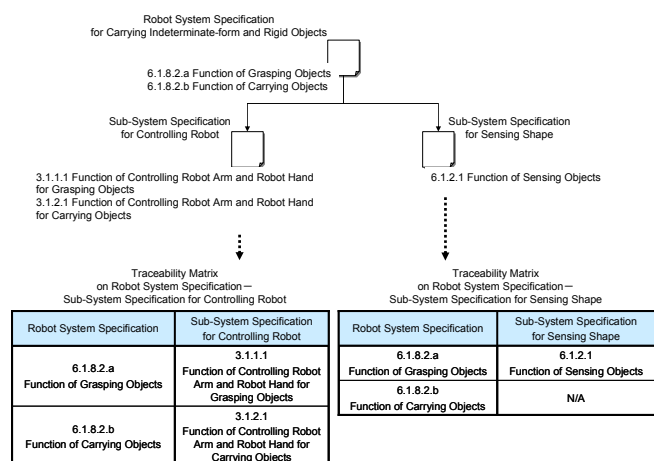


Fig. 8 Traceability Matrix

不定形剛体運搬ロボットシステム仕様 6.1.8.2.a 「Function of Grasping Objects (剛体把持機能)」を基に、ロボット制御サブシステム仕様 3.1.1.1 「Function of Controlling Robot Arm and Robot Hand for Grasping Objects (剛体把持におけるロボットアームおよびロボットハンド制御機能)」および計測制御サブシステム仕様 6.1.2.1 「Function of Sensing Objects (剛体測定機能)」を連携動作する仕様として抽出する。Fig. 9において各サブシステムの仕様における連携動作に関する箇所は、ロボット制御サブシステム仕様および形状測定サブシステム仕様における縞模様部分である。また、構成要素間のインタフェース仕様から連携動作に関する仕様を抽出する際、構成要素の連携動作の仕様における他構成要素とのインタラクション情報を用いる。ここでのインタラクションとは、構成要素間におけるメッセージ送受信などを意味する。構成要素間のインタフェース仕様において、抽出したインタラクション情報に対応する仕様を連携動作に関するインタフェース仕様である。Fig. 9においてサブシステム間のインタフェース仕様における連携動作に関する箇所は、中央のメッセージ部分である。

3.3. 構成要素の連携動作が満たすべき性質の抽出

システム仕様、システム仕様および構成要素の仕様におけるトレーサビリティマトリクスをインプットとして、産業用ロボットの構成要素における連携動作が満たすべき性質を抽出する。本作業のアウトプットは構成要素による連携動作が満たすべき性質である。構成要素による連携動作は、連携動作することにより実現するシステムの仕様を満足する必要がある。連携動作が満たすべき性質は、連携動作の仕様に対応するシステム仕様中に存在する。したがって、トレーサビリティマトリクスを用いて連携動作の仕様からそれに対応するシステム仕様を導出し、導出したシステム仕様から連携動作が満たすべき性質を抽出する。Fig. 8の場合、ロボット制御サブシステム仕様 3.1.1.1 「Function of Controlling Robot Arm and Robot Hand for Grasping Objects (剛体把持におけるロボットアームおよびロボットハンド制御機能)」および計測制

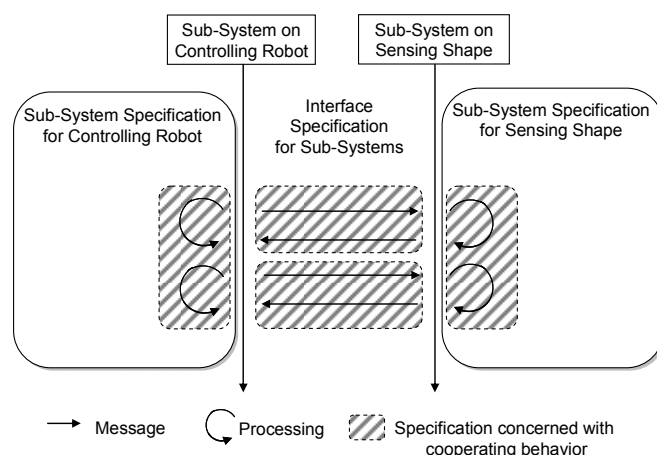


Fig. 9 Specifications concerned with Cooperating Behavior

御サブシステム仕様 6.1.2.1「Function of Sensing Objects (剛体測定機能)」による連携動作が満たすべき性質は、不定形剛体運搬ロボットシステム仕様 6.1.8.2.a「Function of Grasping Objects (剛体把持機能)」から抽出する。

3.4. モデル検査

産業用ロボットにおける構成要素の連携動作に関する仕様および連携動作が満たすべき性質をインプットとしてモデル検査を実施する。モデル検査のアウトプットは、連携動作に関する不整合内容もしくは連携動作が整合している構成要素の仕様および構成要素間のインタフェース仕様である。モデル検査ツールである UPPAAL の表現形式に従い、構成要素の仕様および構成要素間のインタフェース仕様における連携動作に関する仕様をモデル化する。また、UPPAAL の表現形式に従い、連携動作が満たすべき性質からモデル検査の検査式を作成する。作成したモデルおよび検査式を UPPAAL に入力し、モデル検査を実施する。モデル検査の結果、検査式で表す性質をモデルが満足しない場合、UPPAAL から出力される反例を分析し連携動作の不整合内容をアーキテクチャ設計にフィードバックする。フィードバックした連携動作の不整合内容を基に、再度、作業フローに従いアーキテクチャ設計を実施する。モデル検査の結果、検査式で表す性質をモデルが満足する場合、連携動作が整合している構成要素の仕様および構成要素間のインタフェース仕様をリリースする。

4. 産業用ロボットの開発事例

本章では、不定形な剛体を運搬する産業用ロボットの開発において、提案手法を適用した事例を示す。

4.1. 不定形剛体運搬ロボットシステム

不定形剛体運搬ロボットシステムとは、形状・寸法が一定ではない重量のある剛体の把持、運搬、据置を行う産業用ロボットである。不定形剛体運搬ロボットシステムの特徴は、不定形剛体の把持および据置作業に対してつぎに示す強い自律性が求められる点である。不定形剛体運搬ロボットシステムにおいて不定形剛体を把持する領域は限定されているものの、剛体の形状・寸法、剛体を把持する位置、剛体の姿勢は剛体ごとに変化する。システムは剛体を把持するに当たり剛体の形状・寸法、位置、姿勢を正確に知る必要がある。また不定形剛体を据え置く領域は限定されているものの、その領域の中で剛体を据え置く位置が変化する。システムは剛体を据え置くに当たり他の剛体が存在しない位置もしくは剛体が敷き詰められた領域において最も標高が低い位置を正確に知る必要がある。

4.2. 提案手法の適用

不定形剛体運搬ロボットシステムを開発するに当たり、まず開発要求を基にシステムの要求分析を実施した。システムの要求分析を実施することで不定形剛体運搬ロボットシステムのシステム仕様を確定しシステム仕様書を策定した。シス

テム仕様書の策定においては仕様の 1 つ 1 つに対して項番を付与した。これはアーキテクチャ設計の過程で作成するトレーサビリティマトリクスにおいて、システム仕様およびその構成要素の仕様を詳細に対応づける。それにより提案手法における連携動作に関する仕様の抽出および連携動作が満たすべき性質の抽出の精度を向上させるためである。策定したシステム仕様書を基にして提案手法を適用した。本節では提案手法のフローに沿って提案手法の適用内容を述べる。

(1) アーキテクチャ設計

不定形剛体運搬ロボットシステムのシステム仕様を基に、IEEE 1220 に定められるプロセスに従いアーキテクチャ設計を実施した。アーキテクチャ設計の結果、不定形剛体運搬ロボットシステムの仕様をロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様に分解した。Fig. 10 に不定形剛体運搬ロボットシステムのアーキテクチャ設計結果を示す。Fig. 10 にはサブシステムの説明のために、サブシステムを構成するコンポーネントを明示している。

ロボット制御サブシステムはつぎに示すコンポーネントで構成される。

- ロボットハンド
- ロボットアーム
- ロボット動作のプログラミングや緊急停止を行うティーチペンダント
- ロボットハンドを制御するコントローラ
- ロボットアームおよびティーチペンダントを制御するコントローラ
- 運用者が作業指示やシステムのステータス確認を行うコンソール
- ロボット制御サブシステムの統合制御を行うロボット制御計算機

ロボット制御計算機はつぎに述べる形状測定サブシステムとのインタフェースを有する。

形状測定サブシステムはつぎに示すコンポーネントで構成される。

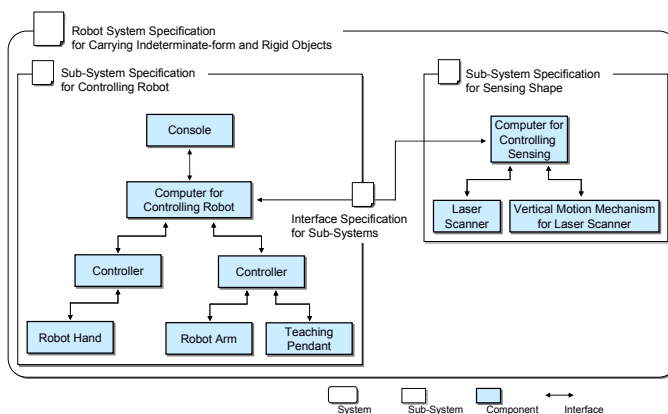


Fig. 10 Results of Architectural Design

- 3次元形状を測定するレーザスキャナ
- レーザスキャナ上下動機構
- レーザスキャナおよびレーザスキャナ上下動機構を制御する測定制御計算機

測定制御計算機はロボット制御サブシステムとのインタフェースを有する。

形状測定サブシステムは剛体を把持する際に剛体の形状・寸法、位置、姿勢を測定する。剛体を据え置く際に据置領域の凹凸状況を測定する。形状測定サブシステムはロボット制御サブシステムに対して測定した情報を送信する。ロボット制御サブシステムは形状測定サブシステムからの情報を基に、不定形剛体の把持、運搬、据置を行う。

アーキテクチャ設計の結果を基にロボット制御サブシステムおよび形状測定サブシステムの仕様書を策定した。それらサブシステム仕様書の策定においても仕様の1つ1つに対して項番を付与した。

また、アーキテクチャ設計の過程においてロボット制御サブシステムおよび形状測定サブシステムに関するトレーサビリティマトリクスを作成した。トレーサビリティマトリクスの作成においては仕様1つ1つに付与した項番単位で不定形剛体運搬ロボットシステムの仕様とロボット制御サブシステムおよび形状測定サブシステムの仕様との対応関係をまとめた。

(2) 構成要素の連携動作に関する仕様の抽出

トレーサビリティマトリクスを基にロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様から連携動作に関する仕様を抽出した。連携動作に関する仕様の抽出について、その一例を示す。つぎに示す仕様はそれぞれの仕様書からの抜粋で設計情報を抽象化している。4章2節(1)で述べたトレーサビリティマトリクスにおいて不定形剛体運搬ロボットシステムの仕様6.1.8.2.aはロボット制御サブシステムの仕様3.1.1.1.aから3.1.1.1.f、形状測定サブシステムの仕様6.1.1.1.aから6.1.3.1.aに対応付けられる。

不定形剛体運搬ロボットシステム仕様

- 6.1.8.2.a 一定の位置に据え置かれている剛体を見出し把持すること。

ロボット制御サブシステム仕様

- 3.1.1.1.a コンソールからの剛体把持要求を待つこと。
3.1.1.1.b 剛体把持要求を受信した場合、剛体測定要求を作成すること。
3.1.1.1.c 作成した剛体測定要求を測定制御サブシステムに対して送信すること。
3.1.1.1.d 剛体測定要求を送信後、測定制御サブシステムからの剛体測定応答の受信待ちを行うこと。
3.1.1.1.e 形状測定サブシステムから剛体測定応答を受信した場合、剛体測定応答の内容を基にロボット

アームおよびロボットハンドを制御し剛体の把持を行うこと。

- 3.1.1.1.f 剛体の把持を完了した場合、剛体把持要求の結果をコンソールに出力すること。

形状測定サブシステム仕様

- 6.1.1.1.1.a ロボット制御サブシステムから剛体測定要求を受信すること。
6.1.1.1.2.a 剛体測定要求を受信した場合、剛体の測定を開始すること。
6.1.2.1.a レーザスキャナを測定開始位置に移動すること。
6.1.2.1.b レーザスキャナを用いて剛体を測定し剛体の標高データを取得すること。
6.1.2.1.c 剛体の標高データの取得時点におけるレーザスキャナの上下位置を取得すること。
6.1.2.1.d 剛体に対する1走査分の標高データの取得が終了した場合、レーザスキャナの上下位置を θ 度進めること。
6.1.2.1.e レーザスキャナの上下位置が ϕ から ϕ 度における標高データを測定すること。
6.1.2.1.f レーザスキャナの上下位置、その位置における剛体の標高データがマトリクスになった測定CSVデータを作成すること。
6.1.3.1.a 測定CSVデータを基に剛体の測定データを算出すること。
6.1.1.2.1.a 剛体測定要求に対する測定処理が正常に終了した場合、ロボット制御サブシステムに対して測定データおよびエラー種別(正常)を送信すること。
6.1.1.2.1.b 剛体測定要求に対する測定処理においてエラーが発生した場合、ロボット制御サブシステムに対して該当するエラー種別を送信すること。

またこれらの連携動作に関する仕様においてサブシステム間のインタラクション情報として剛体測定要求および剛体測定応答に着目する。サブシステム間のインタフェース仕様においてこれらのインタラクション情報に対応する仕様をつぎに示す。仕様に対応する番号はサブシステム間のインタフェース仕様に付与された項番である。

サブシステム間インタフェース仕様

- 4.1 物理インタフェース
4.2.2.1 剛体測定要求/応答シーケンス
4.2.3 データ定義
4.2.4.1 剛体測定要求/応答メッセージフォーマット

連携動作に関する仕様の抽出について、以上に示す手順によりロボット制御サブシステム仕様における仕様78項目から12項目、形状測定サブシステム仕様における仕様58項目から23項目を抽出した。また、サブシステム間のインタフェース仕様における仕様26項目から22項目を抽出した。

(3) 構成要素の連携動作が満たすべき性質の抽出

トレーサビリティマトリクスを基に、不定形剛体運搬ロボットシステムのシステム仕様からロボット制御サブシステムおよび測定制御サブシステムによる連携動作が満たすべき性質を抽出した。連携動作が満たすべき性質の抽出について、その一例を示す。つぎに示す仕様は不定形剛体運搬ロボットシステムの仕様書からの抜粋で設計情報を抽象化している。4章2節(2)に示すロボット制御サブシステムおよび形状測定サブシステムの連携動作に関する仕様は、4章2節(2)で示す不定形剛体運搬ロボットシステムの仕様以外に下記の仕様に対しても対応付けられる。

5.5.2 不定形剛体運搬ロボットシステムを構成する装置間におけるデータ送受信は要求/応答方式を採用すること。

6.3.2.c コンソールからの動作停止要求に対して t_{ms}

以内に不定形剛体運搬ロボットシステムの動作を停止すること。

4章2節(2)で示すロボット制御サブシステムおよび形状測定サブシステムによる連携動作の仕様はこれらのシステム仕様を満たす必要がある。ここで、これらのシステム仕様をロボット制御サブシステムおよび形状測定サブシステムの設計情報で置き換える。これらのシステム仕様から抽出したロボット制御サブシステムおよび形状測定サブシステムによる連携動作が満たすべき性質をつぎに示す。P.1-5 および P.3-3 は、それぞれの性質に付与した番号である。

P.1-5 ロボット制御サブシステムは、形状測定サブシステムに対する剛体測定要求に対し、形状測定サブシステムから剛体測定結果もしくは剛体測定失敗の応答を受信すること。

P.3-3 形状測定サブシステムは、ロボット制御サブシ

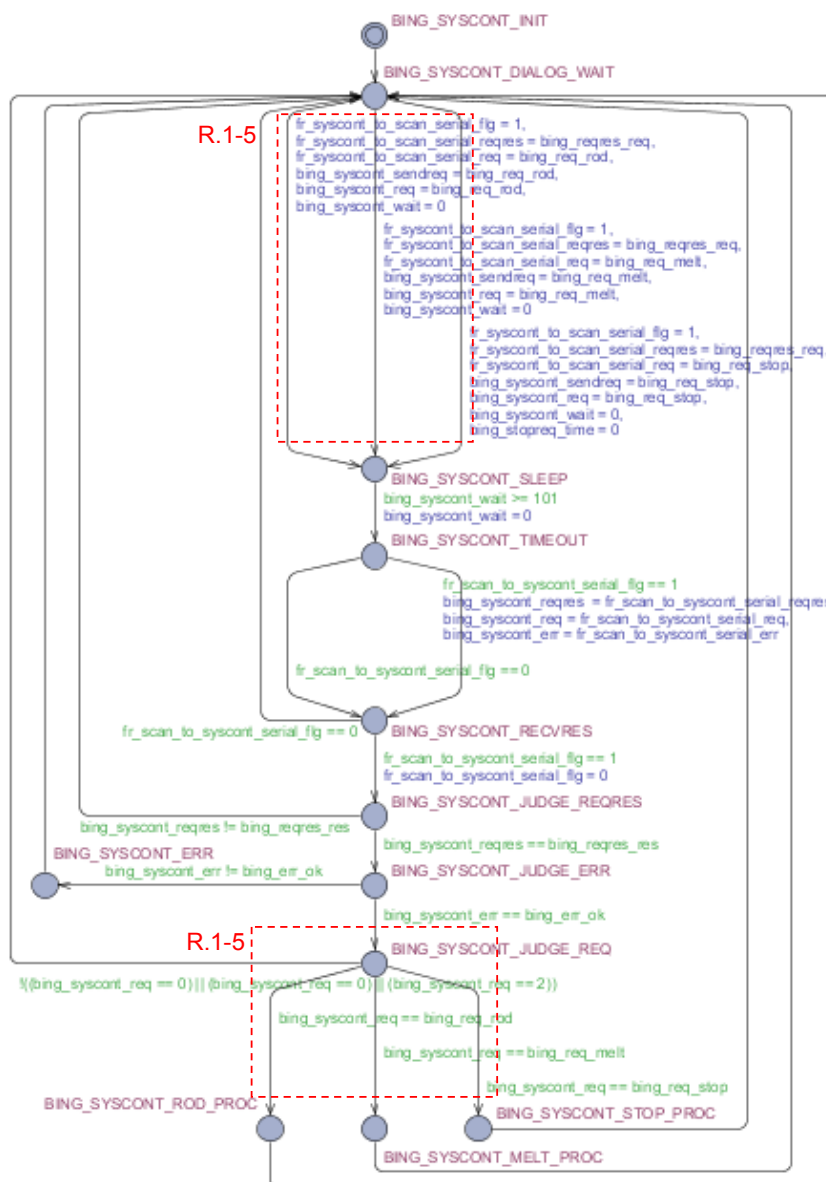


Fig. 11 Cooperating Behavior Model on Sub-System for Controlling Robot

システムによる測定停止要求の送信後 100ms 以内に測定処理を停止すること。

連携動作が満たすべき性質の抽出について、以上に示す手順により 23 項目を抽出した。

(4) モデル検査

ロボット制御サブシステムの仕様および測定制御サブシステムの仕様、サブシステム間のインタフェース仕様から抽出した連携動作に関する仕様を基に、UPPAAL の表現形式に従い連携動作のモデルを作成した。Fig. 11 から Fig. 13 に連携動作のモデルを示す。作成したモデルは連携動作においてロボット制御サブシステムに対応するモデル (Fig. 11)、形状測定サブシステムに対応するモデル (Fig. 12)、サブシステム間のインタフェースに対応するモデル (Fig. 13) である。UPPAAL による連携動作のモデルは連携動作の状態遷移を表現する。状態遷移モデルを作成するにあたり、まず連携動作に関する仕様の中から「～待ち」や「～中」などを識別する。4 章 2 節 (2) に示す形状測定サブシステムの連携動作に関する仕様 6.1.1.2.1.a 「剛体測定要求に対する測定処理が正常に終了した場合、ロボット制御サブシステムに対して測定データおよびエラー種別 (正常) を送信すること」の場合、その仕様には「剛体測定要求に対する測定処理を実施中」という振る舞いが含まれる。これが状態遷移モデルにおける状態となる。つぎに連携動作に関する仕様の中から状態に関連

するイベントを識別する。前述の連携動作に関する仕様の場合、「測定処理の正常終了」がイベントである。これが状態遷移モデルにおける状態の遷移条件となる。そして連携動作に関する仕様の中から状態遷移に関するアクションを識別する。前述の連携動作に関する仕様の場合、「ロボット制御サブシステムに対して測定データおよびエラー種別 (正常) を送信」がアクションである。これが状態遷移モデルにおける状態遷移時に行う手続きとなる。

また、UPPAAL の表現形式に従いロボット制御サブシステムおよび測定制御サブシステムによる連携動作が満たすべき性質を基に検査式を作成した。検査式を作成するにあたり非形式的に表現される満たすべき性質を基に時相論理による正確な検査式を作成する Spec Pattern¹⁷⁾ を適用した。Spec Pattern は満たすべき性質の中に現れるパターンおよびスコープとそれらに対応する時相論理表現とのマッピングを提供している。スコープとは global (常に), before (～の前に), after (～の後に) などモデル検査の対象となるシステムの時系列においてある状態やイベントが生じる範囲を定めたものである。パターンとは absence (ある状態やイベントが指定されたスコープ内で決して生じないこと), universality (ある状態やイベントが指定されたスコープ内で常に生じること), response (指定されたスコープ内で 1 つ目の状態やイベントの発生後に 2 つ目の状態・イベントが常

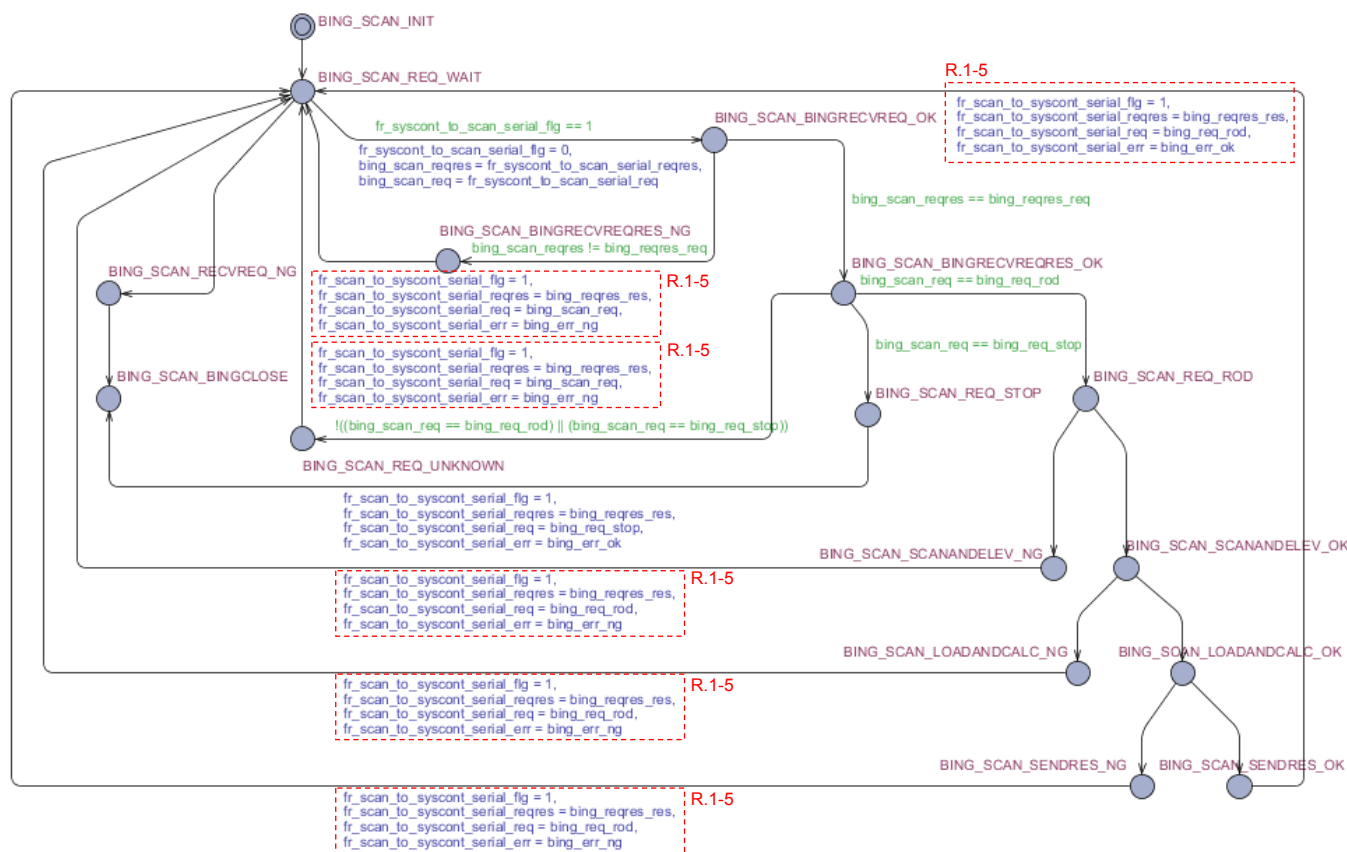


Fig. 12 Cooperating Behavior Model on Sub-System for Sensing Shape

に生じること) などスコープにおける状態やイベントの出現・順序形態を表したものである。Spec Pattern を用いることで時相論理表現に精通していない技術者であっても正確な検査式を作成することができる。ロボット制御サブシステムおよび形状測定サブシステムによる連携動作が満たすべき性質に対して Spec Pattern を適用することで 23 ケースの検査式を作成した。前述の連携動作が満たすべき性質 P.1-5 および P.3-3 に対応する検査式をつぎに示す。F.1-5 および F.3-3 は、それぞれの検査式に付与した番号である。

F.1-5 $A[]$ (bing_syscont_sendreq == bing_req_rod)
 $imply$ (bing_syscont_req == bing_req_rod)

F.3-3 $A[]$ P_BING_SCAN.BING_SCAN_REQ_STOP
 $imply$ (bing_stopreq_time <= 10)

Fig. 11 から Fig. 13 に示す連携動作モデルおよび F.1-5 および F.3-3 を含む 23 ケースの検査式を基に、UPPAAL によるモデル検査を実施した。モデル検査の実施後、UPPAAL により出力された反例を分析した。分析の結果、F.1-5 および F.3-3 の検査式の結果を含む 6 ケースの不整合を検出した。F.1-5 および F.3-3 に対応する連携動作の不整合をつぎに示す。R.1-5 および R.3-3 は、F.1-5 および F.3-3 に対応する連携動作の不整合内容に付与した番号である。R.1-5 については、Fig. 11 および Fig. 12 において連携動作に関する不整合が混入した箇所を点線で示す。

R.1-5 メッセージおよび処理のタイミングによっては、 n 回目に発行された剛体測定要求に対して n 回目より前に発行された剛体測定要求、もしくは据置領域測定要求、もしくは測定停止要求に対する応答が返却される場合がある。

R.3-3 測定停止要求に対して 100ms 以内に測定処理が停止しない場合がある。また、測定停止要求に対して測定サブシステム自体が停止する場合がある。

アーキテクチャ設計に対して R.1-5 および R.3-3 を含む 6 ケースの連携動作の不整合内容をフィードバックした。そして、再度アーキテクチャ設計を実施した。再アーキテクチャ

設計の結果、R.1-5 については形状測定サブシステムおよび測定制御サブシステムの仕様に対してロボット制御サブシステムおよび形状測定サブシステムにおける要求・応答を対応付ける機能を追加した。R.3-3 についてはロボット制御サブシステムからの測定停止要求を常に受信できるように形状測定サブシステムの仕様を変更した。また、測定停止要求を受信した際は直ちに測定を停止し、要求待ち状態に遷移するように形状測定サブシステムの仕様を変更した。その後、連携動作の不整合が解決されたロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様をリリースした。

4.3. 適用結果

不定形剛体運搬ロボットシステムの開発において、提案手法を適用した。その結果、不定形剛体運搬ロボットシステムのシステム仕様から、ロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様を見出した。また、これらの仕様を確定する前に、R.1-5 および R.3-3 に示す連携動作の不整合を検出することができた。その後、連携動作が整合しているロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様を作成することができた。産業用ロボットの開発において提案手法を適用し、産業用ロボットの高信頼化に貢献した。

5. 考察

本章では産業用ロボットに対する提案手法の有効性を示す。また、提案手法の課題を述べる。

5.1. 産業用ロボットに対する提案手法の有効性

4章で示した開発事例を基に、提案手法の有効性を示す。提案手法の有効性を示すにあたり、開発事例における QCD (Quality : 開発対象の品質, Cost : 開発コスト, Delivery : 開発期間) に着目する。

まず、開発対象の品質という観点から提案手法を考察する。今回、不定形剛体運搬ロボットシステムという機能的に複雑な産業用ロボットの開発において提案手法を適用した。その

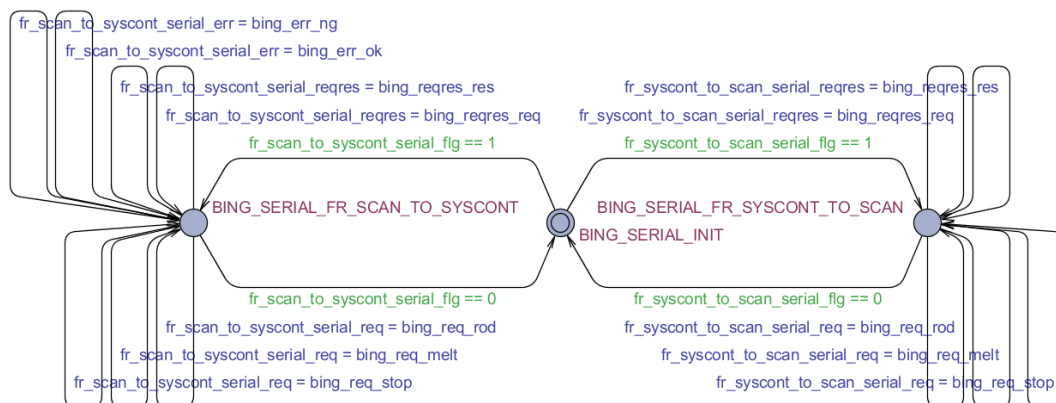


Fig. 13 Cooperating Behavior Model on Interface between Sub-System of Controlling Robot and Sensing Shape

結果、ロボット制御サブシステムおよび形状測定サブシステムにおける連携動作の不整合をアーキテクチャ設計に対してフィードバックし、1回の反復で連携動作が整合しているロボット制御サブシステムの仕様および形状測定サブシステムの仕様、サブシステム間のインタフェース仕様を作成することができた。特に本論文4章2節のR.1-5およびR.3-3に示す連携動作の不整合については、満たすべき性質が成り立つかどうかをしらみつぶしに確認するモデル検査ならではの人手では検出することが困難な不具合といえる。そのような不具合を産業用ロボット開発の上流工程で検出できたことは、提案手法における有効性の表れである。

つぎに、開発コストおよび開発期間という観点から提案手法を考察する。Table. 1に開発事例に対して提案手法を適用した際の工数を示す。本論文ではアーキテクチャ設計を除く作業で要した工数をまとめる。システム開発においてアーキテクチャ設計を含むシステムエンジニアリング手法を適用する効果については、多くの先行研究が存在する^{18)~21)}。これらの先行研究ではシステムエンジニアリング手法を適切に適用することで、システム開発における開発コストの削減および開発期間の短縮が可能であることが示されている。本論文においては、今回の開発事例に対してアーキテクチャ設計手法を適用した際の開発コストの削減および開発期間の短縮の程度を示すことはできない。しかし、前述の通りアーキテクチャ設計において確実な成果が得られたことを考慮した場合、システム開発における開発コストの削減および開発期間の短縮を実現した可能性が高い。また、構成要素の連携動作に関する仕様の抽出については2人時間、構成要素の連携動作が満たすべき性質の抽出については2人時間、モデル検査については12人時間の工数を要した。不定形剛体運搬ロボットシステムを開発するにあたり、7人時間を1人日とした場合に全体で498人日の工数を要した。構成要素の連携動作に関する仕様の抽出、構成要素の連携動作が満たすべき性質の抽出、モデル検査に要した工数の合計14人時間(2人日)は、不定形剛体運搬ロボットシステムの開発全体に要した工数の0.4%程度である。また、サブシステム間における連携動作の不整合について、一般にそれらを検出することが可能な開発フェーズは、システム開発の終盤に実施されるサブシステム同士を組み合わせたシステム試験である。システム試験においてサブシステム間における連携動作の不整合を検出した場合、その改修に多くの改修コストおよび改修期間を要する。文献22)の中でBoehmは、要求仕様の段階で要求の誤りを検出し修正する際のコストを1とした場合、試験においては小規模システムで2、大規模システムで20のコストを要すると分析している。本論文4章2節のR.1-5およびR.3-3に示す連携動作の不整合は、要求仕様を確定する段階においてモデル検査を適用しない限り見逃される可能性がある不具合といえる。システム設計段階において提案手法により検出した不具合を改修するにあたり2人日の工数を要した。

Table. 1 Costs on Applying the Proposed Methodology

| No. | Work | Cost(man-hours) |
|-----|--|-----------------|
| 1 | Architectural Design | — |
| 2 | Extraction of Specification for Cooperating Behavior among Elements | 2 |
| 3 | Identification of Properties for Cooperating Behavior among Elements | 2 |
| 4 | Model Checking | 12 |

システム試験において本不具合が検出されると仮定した場合、それらの改修に40人日の工数を要する可能性がある。これは不定形剛体運搬ロボットシステムの全開発工数の約10%に相当する。したがって、システム開発全体で考えた場合、構成要素の連携動作に関する仕様の抽出、構成要素の連携動作が満たすべき性質の抽出、モデル検査に要する工数は、費用対効果の高い工数であると考えられる。

5.2. 課題

本論文5章1節において、産業用ロボットに対する提案手法の有効性を確認することができた。しかし、提案手法には解決すべき課題がある。提案手法は構成要素の連携動作が整合していることを確認するためにモデル検査を採用している。モデル検査においてモデルの状態数が多い場合、状態の組み合わせが膨大になることでモデル検査が終了しない状態爆発という現象が発生することがある。アーキテクチャ設計の結果、構成要素の連携動作が極端に複雑になると状態爆発によりモデル検査による連携動作の検証が終了しない可能性がある。その場合、構成要素による連携動作が極端に複雑にならないように再度アーキテクチャ設計を実施する方策や連携動作の仕様に関するモデルの状態数を減らす方策などが必要になる。ただしシステム設計段階におけるこれらの工数の増大は、5章1節で述べた通りシステム開発全体では費用対効果の高い工数であると考えられる。また不定形剛体運搬ロボットシステムに対する提案手法の適用ではUPPAALの実行環境としてクロック周波数1.86GHz、メモリ2.96GBを搭載する計算機を使用した。その環境においてFig.11からFig.13に示した連携動作モデルおよび4章2節(4)で述べたF.1-5およびF.3-3を含む23ケースの検査式によるモデル検査を実施した。その結果、それぞれのケースに対する検査は高々1秒以内に終了した。本結果から4章2節(2)の連携動作に関する仕様で示した連携動作の複雑度を有する産業用ロボットの開発においては状態爆発が発生しないことを確認することができた。

6. 関連研究

産業用ロボットの信頼性を向上させる取り組みとして、ロボット開発にモデル検査を適用する方法が提案されている。M. Kim らは Esterel 言語²³⁾ を用いて形式的に定義されたロボットの設計仕様に対し、Esterel 言語に対応したモデル検査ツールである XEVE²⁴⁾ を用いてモデル検査を実施する手法を提案している²⁵⁾。また、N. Trcka らは Chi 言語²⁶⁾ を用いて形式的に定義されたロボットの設計仕様をモデル検査ツール SPIN²⁷⁾ のモデリング言語である PROMELA²⁷⁾ に変換し、SPIN によるモデル検査を実施する手法を提案している²⁸⁾。これらの提案手法は、モデル検査の適用により産業用ロボットにおける信頼性を向上させる可能性がある。しかし、これらの提案手法はロボットの構成要素による連携動作に着目していない。ロボットハンドやロボットアーム、センサなどの構成要素が連携することで満たすべき機能を実現する産業用ロボットにおいては、構成要素による連携動作の信頼性を向上させることが重要である。

本論文における提案手法と同様に、ロボットの構成要素による連携動作に着目した先行研究も存在する。N. C. W. M. Braspenning らは前述の Chi 言語を用いてロボットの各構成要素における設計解をモデル化し、各構成要素のモデルを結合することで Chi システムモデルを作成している。そして、Chi システムモデルをモデル検査ツールである UPPAAL のモデルに変換しモデル検査を実施する手法を提案している²⁹⁾。産業用ロボットの開発において N. C. W. M. Braspenning らの提案手法を適用することで、実プロダクトを結合する前に構成要素による連携動作の不整合を検出できる可能性がある。しかし、この提案手法は構成要素の仕様を基に導出された設計解を対象としている。ロボットのシステム仕様からロボットの構成要素に対して仕様が正しく分割されていることを前提としている。構成要素の実プロダクトを結合する直前に連携動作の不整合を検出するため、開発の上流工程に対する手戻りのインパクトが大きくなる可能性がある。産業用ロボットの開発においては、構成要素による連携動作を確実に設計し、開発の上流工程において連携動作の整合を確認することが重要である。

7. まとめ

本論文では、産業用ロボット開発の上流工程であるシステム設計段階において、産業用ロボットのシステム仕様を連携動作が整合している産業用ロボットにおける構成要素の仕様および構成要素間のインタフェース仕様に分解する開発手法の実例および産業適用の具体例を示した。提案手法の適用結果から産業用ロボットに対する提案手法の有効性を示した。ただし、提案手法には改善の余地もある。今後、本論文で示

した課題の解決を通して、更に品質の高い開発手法の確立を図る。

参考文献

- 1) 日本工業標準調査会：産業用マニピュレーティングロボット用語，JIS B 0134 (2008)
- 2) 社団法人日本ロボット工業会：世界の産業用ロボット稼働台数，社団法人日本ロボット工業会（オンライン），入手先< <http://www.jara.jp/data/dl/kado.pdf> >（参照 2010-03-16）
- 3) 矢内重章：産業用ロボットの現状と将来展望，生産と電気，**61-3**，36/41 (2009)
- 4) 榊原伸介：産業用ロボットの最新技術動向，ロボット，**180**，4/9 (2008)
- 5) 加藤淳，浦郷正隆，狼嘉彰：複雑システムの信頼性を向上させる開発手法：アーキテクチャ設計手法とモデル検査の融合，Synthesiology，**3-3**，197/212 (2010)
- 6) Institute of Electrical and Electronics Engineers (IEEE): IEEE Standard for Application and Management of the Systems Engineering Process, IEEE 1220-2005 (2005)
- 7) K. G. Larsen, P. Pettersson and W. Yi: UPPAAL in a nutshell, International Journal on Software Tools for Technology Transfer, **1**, 134/152 (1997)
- 8) E. M. Clarke, O. Grumberg and D. E. Long: Model Checking and abstraction, ACM Transactions on Programming Languages and Systems, **16-5**, 1512/1542 (1994)
- 9) International Council on Systems Engineering (INCOSE): INCOSE SYSTEMS ENGINEERING HANDBOOK version 3.1, 1.5 of 6, INCOSE, USA (2007)
- 10) T. Doran: IEEE 1220: for practical systems engineering, IEEE Computer, **39-5**, 92/94 (2006)
- 11) J. A. Estefan: Survey of Model-Based Systems Engineering (MBSE) Methodologies, International Council on Systems Engineering (INCOSE) MBSE Focus Group (2007)
- 12) E. M. Clarke and J. M. Wing: Formal methods: state of the art and future directions, ACM Computing Surveys, **28-4**, 626/643 (1996)
- 13) M. Sipser: Introduction to the Theory of Computation, Chapter 1, Course Technology Ptr(SD), The Netherlands (1996)
- 14) R. Alur and D. L. Dill: A theory of timed automata, Theoretical Computer Science, **126-2**, 183/235 (1994)
- 15) E. M. Clarke, E. A. Emerson and A. P. Sistla: Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications, ACM Transactions on Programming Languages and Systems, **8-2**, 244/263 (1986)

- 16) G. Behrmann, A. David, and K. G. Larsen: A tutorial on UPPAAL, Proceedings of Formal Methods for the Design of Real-Time Systems(SFM-RT 2004), **3185** of Lecture Notes in Computer Science, 200/236 (2004)
- 17) M. B. Dwyer, G. S. Avrunin and C. Corbett: Pattern in Property Specification for Finite-State Verification, Proceedings of the 21th International Conference on Software Engineering, 411/420 (1999)
- 18) J. P. Elm: A study of Systems Engineering Effectiveness – Initial Results, Proceedings of the Systems Conference 2008 2nd Annual IEEE, 1/7 (2008)
- 19) B. Boehm, R. Valerdi and E. Honour: The ROI of Systems Engineering: Some Quantitative Results for Software-Intensive Systems, Systems Engineering, **11-3**, 221/234 (2008)
- 20) E. C. Honour: Understanding the Value of Systems Engineering, Proceedings of the INCOSE International Symposium, 1-16 (2004)
- 21) A. K. Kludze: The Impact of Systems Engineering on Complex Systems, Proceedings of Conference on Systems Engineering Research (2004)
- 22) B. W. Boehm: Software Engineering Economics, 38-40, PRENTICE-HALL, USA (1981)
- 23) G. Berry and G. Gonthier: The Esterel synchronous programming language: Design, semantics, implementation, Science of Computer Programming **19-2**, 87/152 (1992)
- 24) A. Bouali: XEVE, an ESTEREL Verification Environment, Proceedings of the 10th International Conference on Computer-Aided Verification (CAV'98), **1427** of Lecture Notes in Computer Science, 500/504 (1998)
- 25) M. Kim, K. C. Kang and H.Lee: Formal Verification of Robot Movements – a Case study on Home Service Robot SHR100, Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA), 4739/4744 (2005)
- 26) D. A. Ban Beek and J. E. Rooda: LANGUAGES AND APPLICATIONS IN HYBRID MODELLING AND SIMULATION: POSITIONING OF CHI, Control Engineering Practice, **8-1**, 81/91 (2000)
- 27) G. J. Holzmann: The Model Checker SPIN, IEEE Transactions on Software Engineering, **23-5**, 279/295 (1997)
- 28) N. Trecka: Verifying Chi Models of Industrial Systems with Spin, Proceedings of the 8th International Conference on Formal Engineering Methods (ICFEM'06), **4260** of Lecture Notes in Computer Science, 132/148 (2006)
- 29) N. C. W. M. Braspenning, E.M. Bortnik, J. M. van de Mortel-Fronczak and J. E. Rooda: Model-based system analysis using Chi and Uppaal: An industrial case study, Computers in Industry, **59-1**, 41/54 (2008)

著者紹介

加藤 淳



2000年熊本大学大学院自然科学研究科電気システム専攻修了。同年より電機メーカーにおいて、ユビキタス環境における小型ネットワークデバイスの研究・開発業務などに従事。2006年より宇宙業界において、宇宙機ソフトウェアの独立評価業務に従事。また、慶應義塾大学大学院システムデザイン・マネジメント研究科において、システムエンジニアリングに関する研究に取り組む。2009年情報処理学会研究賞受賞。情報処理学会会員。

浦郷 正隆



1998年東京工業大学大学院理工学研究科機械物理学専攻博士課程修了。博士(工学)。同年より東京工業大学で助手を務める。2008年より慶應義塾大学大学院システムデザイン・マネジメント研究科准教授となる。システムズエンジニアリングおよび工学問題のコンピュータモデリング、数値計算に関する研究に取り組む。日本機械学会会員。INCOSE 会員。日本計算数理工学会会員。

狼 嘉彰 (正会員)



1968年東京工業大学大学院理工学研究科電気工学専攻修了。工学博士。NASA国際フェロー、東京工業大学教授、慶應義塾大学教授、宇宙開発事業団技術研究本部研究総監を経て、2008年より慶應義塾大学大学院システムデザイン・マネジメント研究科委員長を務める。複雑システムのダイナミクス・制御および戦略的システムエンジニアリングに関する研究に取り組む。計測自動制御学会産業論文 Associate Editor。日本機械学会フェロー。INCOSEフェロー。日本航空宇宙学会会員。IEEE 会員。AIAA 会員。ASME 会員。