

MBSE ツールを用いた AUTOSAR プラットフォームの ビジュアルモデリングと設計手法

高田光隆¹ 松原豊¹

概要：複雑・大規模化している車両開発において、車載ソフトウェアプラットフォームである AUTOSAR を用いたソフトウェア開発が取り組まれている。AUTOSAR は多くのコンポーネントで構成されており、車両システムやコンポーネント間の関係はあらかじめ決められたモデル情報を用いて設計・記述される。モデル情報のビジュアル化は抽象度が高い車両レベルやシステム設計を把握するために有用であるが、ビジュアルツールが高価であり、詳細設計で利用する際に実際の通信経路をビジュアルで表現するには情報量が過ぎるため、表形式による表示などすべてモデルによるビジュアル化を望まないという側面もある。本研究では Model-based Systems Engineering(MBSE)ツールで記述できるビジュアルモデリング情報から AUTOSAR の設計情報への関連性を整理し、車両・システム設計情報への具体的な変換方法を検討した。また検討の検証のため MBSE ツールで記述したモデル情報から AUTOSAR モデルへのモデル変換ツールを開発し確認を行った。

キーワード：AUTOSAR, モデリング, MBSE, Capella, モデル変換

1. はじめに

複雑・大規模化している車両開発において、車載ソフトウェアプラットフォームである AUTOSAR[1]を用いたソフトウェア開発が欧州や日本などで取り組まれている。AUTOSAR は多くのコンポーネント群で構成されており、車両システムやコンポーネント間の関係は決められたモデル情報を用いて設計・記述される。モデル情報のビジュアル化は抽象度が高い車両レベルやシステム設計を把握するために有用である。

しかし AUTOSAR 公式モデリングツールである Artop[2]は XML によるモデル情報の記述やツリー表示しか持たないため、ビジュアルモデリング機能はサポートされていない。また既存のビジュアルモデリングツールを AUTOSAR のモデル設計に利用する場合、ツールと AUTOSAR モデル間の関連性が定義されていないため、ビジュアルモデリングツール毎に記述レベルや方法が異なるなどの課題がある。

本研究では Model-based systems engineering(MBSE)ツールで記述できるビジュアルモデリングに着目して、MBSE ツールと AUTOSAR のモデル情報の関連性を整理し、MBSE ツールによるビジュアルモデリングから車両・システム設計情報への具体的な変換方法を検討した。また検討結果をもとに、MBSE ツールから AUTOSAR へのモデル変換ツールを開発し、単純な通信アプリケーションを対象にモデル変換の妥当性を確認した。

表 1 オープンソースライセンスの MBSE ツール

ツール名	主体組織	言語
Capella	Eclipse, Polarsys	独自 (Arcadia)
Modelio	Modeliosoft	SysML
Papyrus	Eclipse, Papyrus	SysML
SysML Designer	Obeo	SyML

* 1 名古屋大学
Nagoya University

2. MBSE ツールによるビジュアルモデリング

ビジュアルモデリングを行える MBSE ツールは商用やオープンソースのライセンスを問わずある(表 1)が、本研究ではツールの改造が容易に行えるオープンソースライセンスのツールであり、SysML から派生したモデル記述が可能な Capella[3]を採用した。SysML によるシステム設計のガイドライン[4]は存在するが、さまざまなシステムドメインを対応するため AUTOSAR の実装プラットフォームで使用する場合にはモデル情報記述の拡張が必要になる[5]。

2.1 Capella

Capella は MBSE のためのオープンソースソリューションでシステム、ハードウェア、ソフトウェアアーキテクチャをグラフィカルにモデリングするためのプロセスとツールを提供している。Capella は主に航空、宇宙、輸送、通信、セキュリティ、自動車などの組込みシステム開発において複雑かつ安全性が重要なシステムのモデリングに使用されている。Capella ではシステムを理解するために 4 つのアーキテクチャ階層(表 2)を設け、各階層でシステムを理解するために必要な情報を記載する方法論(Arcadia Method)[6]を採用している。

表 2 Capella のアーキテクチャ階層

アーキテクチャ階層	内容
Operational Analysis	システムの利用者が何を達成する必要があるのか
Functional & Non Functional Need	ユーザが必要とするシステム機能
Logical Architecture	ユーザの期待に応えるために、システムがどのように機能するか
Physical Architecture	システムをどのように開発・構築するか

2.2 Capella Logical Architecture

Capella Logical Architecture (Capella LA) ではシステムの技術や実装検討とは別に、ユーザの期待に応えるためシステムがどのように機能するかといった、システム内部の論理的構成要素および振る舞いを特定することを記述の目的としている。システムの内部機能分析を行うために Capella LA では次のようなモデル要素のコンセプトを用意してモデリングの論理設計記述を行っている (表 3)。

Capella LA によるモデル情報の記述は上流のシステムレベルの情報から記載する Logical Component Breakdown diagram(LCBD)と直接アーキテクチャ図を作成する Logical Architecture Blank(LAB)の方法がある。本研究では Capella と AUTOSAR のモデル情報に関する関連性を明らかにするのが目的のため、より Capella での記述が柔軟にできる LAB を使ってアーキテクチャ図を作成する方法をとった。

Capella LA では Logical Component や Logical Actor は自身の要素を子要素として構成する入れ子構造を表現することができるが Logical Function は自身の要素を入れ子構造で表現することはできず 1 階層のみの記述となる (図 1)。

表 3 Capella LA の主なコンセプトモデル要素

要素名	内容
Logical Component	システム内の構成要素で、他の Logical Component や外部 Actor と相互作用する構造的なポートを持つ
Logical Actor	システムの外部にあり、システムと相互作用するあらゆる要素
Logical Function	Logical Component または Logical Actor が提供する動作またはサービス
Functional Exchange	2つの Logical Ports を結び、2つの Logical Function 間で行われる情報の一方向性の交換
Component Exchange	Logical Component, Logical Actor 間で行われる情報の交換
Logical Scenario	Logical Component や Logical Actor とのシーケンスで表現する
Functional Chain	特定の経路を指定することができるモデル要素

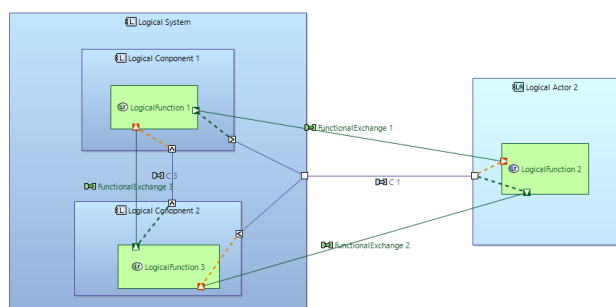


図 1 Capella Logical Architecture による記述例

2.3 Capella Physical Architecture

Capella Physical Architecture (Capella PA) ではシステムをどのように開発・構築するかに着目し、Capella LA からコンセプトモデル要素を追加することで、Capella PA において実装設計の記述を可能としている (表 4)。

Capella PA も LA と同様に上流アーキテクチャからモデル情報を移行する方法である Physical Component Breakdown diagram(PCBD)による記述方法があるが、Capella LA と同様に直接モデリング記述を行う Physical Architecture Blank (PAB) による方法で調査を行うことにした。

Capella PA では Arcadia のルールに従って、Node Components と Behavior Components といった物理コンポーネント要素が 2 種類ある。Behavior Components はシステムのコンポーネントであり、他のコンポーネントやアクターと相互作用することでシステムに割り当てられた機能の一部を実行する。Node Components は Behavior Components へ物理的なリソースを提供するコンポーネントとなっており、Capella LA で記述をしたコンポーネントは Capella PA で振る舞いと物理的なノードとして具体化した記述が可能となる (図 2)。

表 4 Capella PA で追加したコンセプトモデル要素

要素名	内容
Behavior Physical Component	物理的な機能を持ち、システム動作の一部を担う物理コンポーネント
Node Physical Component	1 つ以上の Behavior Physical component に必要な物理的リソースを提供する物理コンポーネント
Physical Port	Behavior / Node Component に属する無指向性のポート
Physical Link	Physical Port 間の無指向性の物理的接続 (例: イーサネット, USB ケーブル)
Physical Path	複数の Physical Link を繋いで Component Exchange が複数の Node を経由する経路

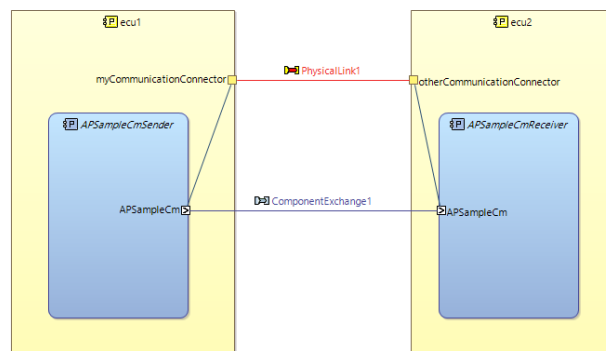


図 2 Capella PA によるコンポーネント要素の記述例

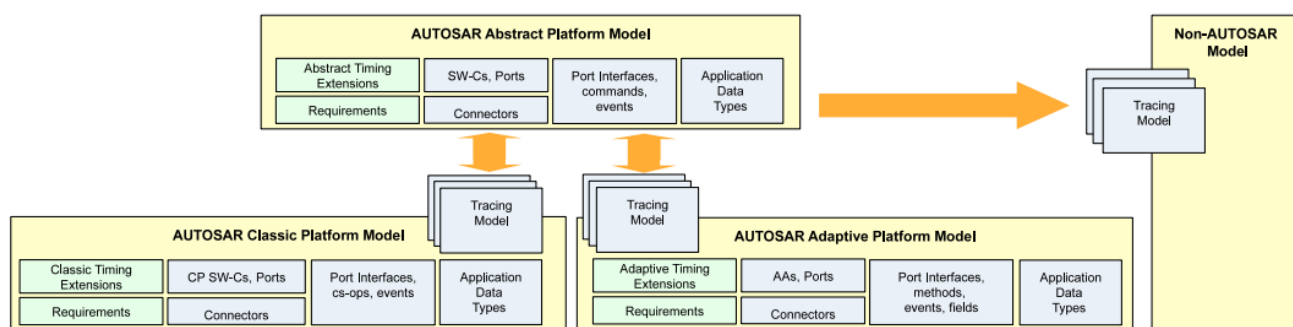


図 3 AUTOSAR Abstract Platform と他のプラットフォームの関係

3. AUTOSAR プラットフォーム

AUTOSAR のモデリングは UML コンポーネント図の構成要素をベースに各プラットフォームの利用に応じて適切な要素の追加や制約を行って記述を実現している。AUTOSAR のモデリングは XML による記述(ARXML)で表現されているが, Artop は XML ツリーの表示・編集機能と関連するモデル情報へのアシスト機能しかない。コンポーネント図のようなビジュアルモデリングによるソフトウェア設計を行うことは、複雑なコンポーネント間の関係性が見える化し理解しやすくなる。以下では AUTOSAR 各プラットフォームの概要とビジュアルモデリングに必要な構成要素について説明する。

3.1 AUTOSAR Abstract Platform

AUTOSAR Abstract Platform(AUTOSAR XP)[7]は車載システムに対するより抽象度の高いレベルのソフトウェアビューを実現するため、実装ソフトウェアプラットフォームである AUTOSAR Classic Platform (AUTOSAR CP)や AUTOSAR Adaptive Platform (AUTOSAR AP)より汎用的なプラットフォーム記述として位置づけられている。

AUTOSAR XP 仕様策定の時系列をみると, AUTOSAR CP および AP の後になっており, AUTOSAR の Foundation (共通基盤)の一つとして AUTOSAR CP や AP からは独立したプラットフォームとなっている。そのためトップダウン的な開発プロセスにおいて AUTOSAR CP, AP を採用する際に必ず AUTOSAR XP を経由する必要はない。AUTOSAR XP は AUTOSAR CP や AP, 他の車載プラットフォームへの相互移行を行う際の共通的なモデルビューを提供することを目的としている (図 3)。

AUTOSAR XP では次のような AUTOSAR CP, AP のモデル要素を抽象化したモデリングのビュー環境を実現している。

(1) Components (コンポーネント)

ある機能を実現する単位のこと、ソフトウェアコンポーネントと同義。複数のソフトウェアコンポーネントを合成したコンポジションソフトウェアコンポーネントから成る。

(2) Port-interfaces (ポートインタフェース)

コンポーネント間をつなぐ際の窓口となるもの。AUTOSAR XP では通信の方向によって要求ポートまたは提供ポートのどちらかを記載することになっており、要求・提供の両方をサポートするポートインタフェースの記述はできない。AUTOSAR CP, AP ではポートインタフェースに特定の型(種類)を設けているが, AUTOSAR XP では通信の方向だけが設定でき、そのサブクラスにおいて具体的な型やデータタイプを指定する。

(3) Connectors (コネクタ)

AUTOSAR XP ではコネクタはポートを結ぶ結線としてビジュアルに表現されている。XP では特に具体的な意味を持たせず、下位のプラットフォームにその使い方をまかせている。

(4) DataType (データタイプ)

ポートインタフェースを通じて流れる情報の型。ツールのビジュアルモデリングとしては表現されないが、ポートインタフェースで扱うデータを決定するためにモデル情報として存在する。

3.2 AUTOSAR Classic Platform

AUTOSAR CP [8]は従来からの車載組込み制御向けソフトウェアプラットフォームの仕様で 2003 年から仕様策定が行われている。上流開発の車両システム設計から下流開発のソフトウェアコンポーネントのパラメータ設定までの各開発工程を ARXML で記述する。ARXML でコンポーネントやパラメータ情報を受け渡しすることで、多くの開発工程での情報の標準化を図っている。また AUTOSAR では開発工程ごとの入出力情報をモデル化し ARXML で受け渡しするとともに、各開発工程においてどのようなツールを使うことを求めるかといった方法論まで仕様化[9]されているのが特徴である。

(1) Components (コンポーネント)

システムを構築する際に使用する中心的な構造要素。コンポーネントは明確に定義されたポートがあり、それを通じてコンポーネントが他のコンポーネントと相互作用する。

(2) Ports (ポート)

ポートはコンポーネント間の端点である。コンポーネント間のデータの流れる方向に応じていくつかのポート種類がある。

(3) Port-Interfaces (ポートインタフェース)

コンポーネントのポートを接続する際の規約であり、規約の種類に応じて、Client-server や Sender-receiver といった種類がある。同じポートインタフェース同士かつポートの方向の記述が正しくない場合、モデル情報を記述するツールや実行ファイルを生成する工程でエラーとなる。

(4) Connectors (コネクタ)

システム設計などコンポーネントのポート間の結線として表現される。

3.3 AUTOSAR Adaptive Platform

1 億行を超えるとされる車載ソフトウェア開発のソースコードにおいて、AUTOSAR CP のようにソフトウェアのモジュール構造による開発の効率化だけでは、より高度で複雑な自動運転や動的なソフトウェアの変更などのニーズに対応することが難しい。そのため AUTOSAR ではそれらのニーズに対応するためのプラットフォームとして AUTOSAR AP [10]を策定している。

基本となるソフトウェアプラットフォームのモデリングは AUTOSAR CP と変わらないが、AUTOSAR AP のモデリングに影響をあたえるコンセプトとして、Service-oriented-architecture (SOA)、アジャイル開発、Manifestなどを挙げることができる。

SOA ではシステムはサービスの集合で構成されると考えアプリケーションは必要に応じてサービスを利用するという概念に基づいている。AUTOSAR CP のシステム設計は VFB(Virtual Functional Bus)[11]と呼ばれる仮想バスのコンセプトを用いて機能モジュールを容易に付け替え可能としている。ところが実装段階ではモデル要素が固定され、設計変更で上流へ戻る場合、再設計後の影響範囲が大きく実装設計者がシステム全体を理解する必要があるなど作業負荷が高い。SOA を採用した AUTOSAR AP ではコンポーネント間の結合度をより柔軟にする実装方法を採用することで、下流工程の変更による影響度が AUTOSAR CP より少なくできるというメリットがある。

アジャイル開発は SOA を採用した AUTOSAR AP の開発に適しており、例えば自動運転技術を確立するための設計、開発、テストと言った開発のサイクルをシミュレーションや実機を用いて早くフィードバックして繰り返し行えるようにするための開発手法として積極的に導入している。

Manifest は AUTOSAR AP 構成をサポートするために作成され、ソフトウェアの実行コードを含む成果物と組み合わせ導入され、AUTOSAR モデル記述の一部として表される。Manifest はソフトウェアのコンパイル時や実行時に使用する設定ファイルとして実際には使用される。

4. Capella から AUTOSAR へのモデル変換

Capella アーキテクチャや AUTOSAR の各プラットフォームにおけるモデル要素を踏まえ、Capella モデルから AUTOSAR モデルへのモデル変換について検討を行った。検討対象のユースケースとして AUTOSAR AP の教育で利用している単純な通信アプリケーションを用いることとした。対象アプリケーションのモデルに関する情報はアプリケーション概要図と AUTOSAR AP の ARXML ファイルだけである。

4.1 Capella LA から AUTOSAR XP へのモデル変換

AUTOSAR でも抽象度の高いプラットフォームである AUTOSAR XP へのモデル変換への検討を行った。

初めに、Capella LA で通信アプリケーションのモデリングを行った(図4)。モデリングの制約としてコンセプト要素のうち、Functions 要素(Logical Function, Functional Exchange, Functional Chain)は用いず Components 要素だけを使用し、設定情報の記述では Capella の標準エディタ機能以外のプロパティ情報は追加をしないこととした。

その後、モデリングで記述した要素を AUTOSAR XP モデルと対比した。検討ではすべてのモデル要素に対応することとはせず、AUTOSAR モデルのコンセプト要素のうちコンポーネント、ポート、ポートインタフェース、コネクタ要素を変換対象とした(表5)。

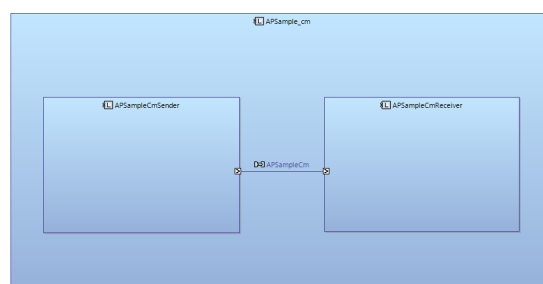


図4 Capella LA による通信アプリケーション

表5 Capella LA と AUTOSAR XP のモデル要素対応

Capella LA モデル	AUTOSAR モデルコンセ プト要素	AUTOSAR XP モデル
Logical Component	Components	RootSwCompositionPrototype
Logical Component	Components	SwComponentPrototype
In Flow Port	Ports	RPortPrototype
Out Flow Port	Ports	PPortPrototype
Interface	Port-Interfaces	CompositeInterface
Component Exchange	Connectors	AssemblySwConnector

モデル要素の対応表（表 5）を元にモデル変換ツールを作成した。変換ツールは Capella のモデル情報を入力として、AUTOSAR のモデル記述である ARXML ファイルを出力とした。Capella および AUTOSAR モデリングツールである Artop は Eclipse をベースに作られていることから、モデル変換ツールは Eclipse のプラグインとして開発した。

4.2 Capella PA から AUTOSAR AP へのモデル変換

次に AUTOSAR の実装プラットフォームである AUTOSAR AP へのモデル変換について検討を行った。

AUTOSAR AP は XP に比べて表現するモデル要素が多い。さらに Capella PA ではコンポーネント要素を Node と Behavior の 2 種類にわけているので、その分類基準について調査した。AUTOSAR AP の方法論[12]では、Vehicle Software Architecture から Machine への対応に関する記述があり、そこではシステムのハードウェア構成である Vehicle Hardware Architecture として以下が構成されるとあった。

- AP Machine および ECU-Instance
- SOME/IP などの通信クラスタ
- 無線通信用コネクタ

そこでこれらの構成要素に関しては Node コンポーネント要素として分類することとし、それ以外の情報は Behavior コンポーネント要素として分けることとした。

AUTOSAR XP へのモデル変換と同様に通信アプリケーションを Capella PA でモデリングするところからはじめた。モデリングの時点ではコンポーネントの分類を考慮しながらもそれ以外の AUTOSAR AP の対応関係については気にせず Capella PA のコンセプト要素が過不足なく記載するようにビジュアルモデリングを行った（図 5）。

その後、通信アプリケーションの Capella PA と AUTOSAR AP のモデル要素の関係性について検討した。AUTOSAR AP モデルである Machine と対になる MachineDesign は事前調査で対応済みであるが、それ以外は通信アプリケーションで使った AUTOSAR メタモデルを抽出し、Capella PA との関係についてまとめた（表 6）。

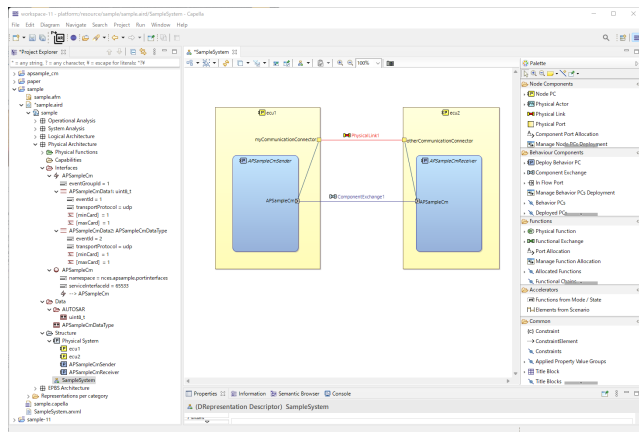


図 5 Capella PA による通信アプリケーション

表 6 Capella PA と AUTOSAR AP のモデル対応（抜粋）

Capella PA モデル	AUTOSAR モ デルコンセ プト要素	AUTOSAR AP モデル
Node PC	Components	Machine, MachineDesign
Physical Link	Connectors	EthernetCommunicationConne ctor
Physical Port	Ports	coomunicationConnector, serviceDiscoverConfig
Component Port Allocation	Connectors	SomeipServiceInstanceToMach ineMapping
Deploy Behavior PC	Components	contextRootSwComponentProt otype, AdaptiveApplicationSwCompo nentType, Executable, Process, ProcessDesign
Component Exchange	Port-Interfaces	event, eventGroup, eventDeployment
In Flow Port	Ports	RequiredSomeipServiceInstanc e
Out Flow Port	Ports	ProvidedSomeipServiceInstanc e

検討した結果、多くのモデル要素は Capella PA と AUTOSAR AP で 1:1 対応できることがわかった。そして Capella PA から AUTOSAR AP へモデル変換できない要素としては、ネットワーク設定情報や SOA のサービス ID など Capella PA が用意しているプロパティ定義には無い要素であることがわかった。

そこで対応できないプロパティ定義については、Capella PA で拡張プロパティの記述を追加することで AUTOSAR AP へモデル変換が可能であると検討し、モデル変換ツールを開発した。

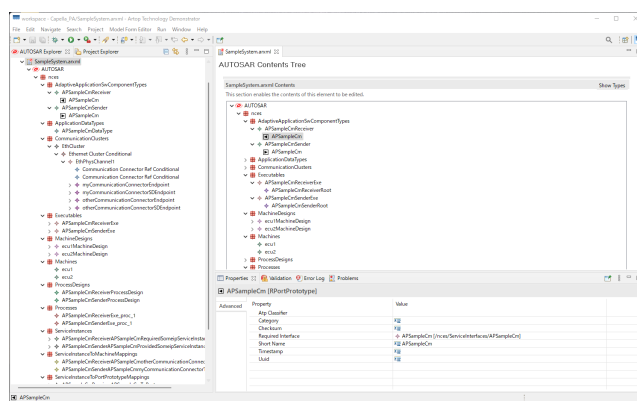


図 6 モデル変換した ARXML ファイル

4.3 Capella PA から AUTOSAR AP へのモデル変換の評価

AUTOSAR AP ではアジャイル開発の一環としてシステム機能を確認するための仕様と確認プログラムおよびその開発環境が含まれる AP Demonstrator (APD) が用意されている。

本研究の評価として、APD を用いて実際に通信アプリケーションが動作できるかということで妥当性を確認した。

Capella PA でモデリングした通信アプリケーションをモデル変換ツールで AUTOSAR AP の ARXML ファイル形式へ変換し、Artop でファイルを読み込み、スキーマにエラーが無いことを確認した (図 6)。その後、変換した ARXML ファイルは通信アプリケーションのソースコードと組み合わせ、APD を用いてコンパイルを行い、対象環境で通信アプリケーションが動作することを確認した。

今回モデル変換のユースケース対象とした通信アプリケーションは AUTOSAR AP を学ぶため単純な機能構成となっている。ARXML のスキーマ [13]によると AUTOSAR AP に関するメタモデル要素の定義が 374 あり、本研究で開発したモデル変換ツールではそのうち 15% の 58 モデル要素を対応することができた。しかし検討ユースケースでは限られたモデル情報のみの対応となっている。そのためポートインタフェースは Event 通信を対応したが、Field や Method 通信は対応していない。また通信プロトコルについても SOME/IP のみで、CAN や DDS といった他の通信プロトコルへの対応も今後の課題となる。

5. まとめ

AUTOSAR のビジュアルモデリングは教育や設計初期段階などシステムの概要を理解することに適しているが、実際には利用可能な簡易ビジュアルモデリングツールは存在していなかった。そこで MBSE ツールのビジュアルモデリングに着目し、Capella LA と PA で記述した単純な通信アプリケーションのモデリングから AUTOSAR XP と AP のプラットフォームへのモデル変換を検討した。

そして検証のためのモデル変換ツールを開発し、Capella PA から AUTOSAR AP へモデル変換した情報を通信アプリケーションで使用し、検討したモデル変換の妥当性を確認した。

今後は本開発ツールを用いたビジュアルモデリングの記述例を増やしつつ、AUTOSAR AP の通信方式や通信プロトコルなどモデル変換の対象要素を拡充するとともに AUTOSAR 方法論に適した Capella LA から Capella PA へのモデルの自動移行や他のプラットフォームへの対応などを検討していきたい。

参考文献

- [1]“AUTOSAR”. <http://www.autosar.org/>, (参照 2022-02-04).
- [2]“Artop”. <http://www.artop.org/>, (参照 2022-02-04).
- [3]“Eclipse Capella”. <https://www.eclipse.org/capella/>, (参照 2022-02-04).
- [4]DEOS. 対象システムの SysML モデリングガイド, 2014.
- [5]Andreas Korf.. AUTOSAR and SysML – A Natural Fit?. Conference ERTS’06. 2006.
- [6]Jean-Luc Voirin. Model-based System and Architecture Engineering with the Arcadia Method. Elsevier. 2017.
- [7]AUTOSAR. Specification of Abstract Platform. AUTOSAR FO R21-11, 2021.
- [8]AUTOSAR. Layered Software Architecture. AUTOSAR CP R21-11, 2021.
- [9]AUTOSAR. Methodology Classic Platform. AUTOSAR CP R21-11, 2021.
- [10]AUTOSAR. Explanation of Adaptive Platform Design. AUTOSAR CP R21-11, 2021.
- [11]AUTOSAR. Virtual Functional Bus. AUTOSAR CP R21-11, 2021.
- [12]AUTOSAR. Methodology for Adaptive Platform R21-11, 2021.
- [13]AUTOSAR. Meta Model-generated XML Schema. AUTOSAR FO R21-11, 2021.