

ソフトウェア開発のモデリングは どう進化したら効果的なのか？ 過去を振り返りながら、皆で議論提案してみよう！

2016-8-26

第18回 組込みシステム技術に関するサマーワークショップSWEST18

The logo for AISIN comCruise, featuring the word "AISIN" in a large, bold, white sans-serif font, followed by "comCruise" in a smaller, white sans-serif font. The background of the slide is a photograph of a red and white cruise ship sailing on a blue sea under a blue sky with light clouds.

AISIN comCruise

ソフトウェア開発のモデリングは どう進化したら効果的なのか？

過去を振り返りながら、皆で議論提案してみよう！

仕様書などシステム記述やシステムズエンジニアリング目的の sysML、制御/シミュレーションに注目した Matlab/Simulink の MBD、ソフトウェア設計表現のしやすさに注目した UML の MDD などモデルベース設計は着実に主流になっています。

一方でこれらの標準モデリングの基本はというと、10年前に提案されてから大きく変わっていません。

パラダイムシフトは10年毎に起きていて、技術変化がどんどん早くなっている現在、いろいろなモデリングの補完/新規提案も始まっています。

考えることは世界同じで、タイミングよく sysML が 2.0 への改定に OMG が昨年後半から動き出しました。

① それらの経緯と近況を俯瞰しましょう

1-1 モデリングの歴史を振り返ります

1-2 sysML 2.0 の提案や課題認識をみてみましょう

1-3 その他で最近提案されているモデリングを俯瞰してみましょう

② 皆さんと背景や課題(モデリングの長所、短所、限界)を考えてみます

③ これからはこれが良い！と思うモデリングを皆さんで議論したいと思います

広島大学/工学部電子工学科卒 アイシン精機入社後2014年 アイシン・コムクルーズへ

- 4ビットマイコンの時代から自動変速機やアクティブ・サスペンション,ABS,ESC等のコンピュータ、関連するハード、IC設計、実装技術の開発・製品化に従事。その後開発比重の高くなったソフトウェア課題に取り組む。機能安全は、その延長で取り組んで現在に至る。現在はクリチカルシステムの効率的開発やASEAN連携が関心事。
- 個人的には、パソコンオタク、コンパイラマニア、古代歴史マニア
- 九州工大 産官学連携講座 非常勤講師 ('08~)
- 新エネルギー・産業技術総合開発機構NEDO 新技術調査委員 :ITベンチャー応援団('14~)
- IPA高信頼性部会委員('14~)、ET/IoT2015/16実行委員('15~)、オートモティブソフトウェアフロンティアフォーラム実行委員('15~)
- NEDO 生活支援ロボット実用化プロジェクト第2期 アイシンコンソ 准代表&研究員 ('11~'13)
- 自動車技術会 電子・電装部会/電子機能対応分科会 ISO26262規格審議委員 ('8-'13)
- 経済産業省委託 海外自動車電子化調査WGリーダー等
 - 「平成18年度 Jaspar委託調査 Autosar周辺調査」
 - 「平成19-24年度 (財)JARI ITS規格化事業/ITS自動車の電子化に係る欧州調査」
- 地域コンソーシアムプロジェクト活動参加 (経済産業省助成)
 - 自動車統合制御用組込みOSの開発 '05~
 - 機能安全対応自動車制御用プラットフォームの開発 '06~
 - 形式的仕様記述を用いた高信頼ソフト開発プロセス研究とツール開発 '10~
 - 故障未然防衛機能を有す高信頼ソフトウェアプラットフォームの開発 '10~
 - 高度IT 融合社会を支える次世代自動車用セキュリティ・ゲートウェイ・ECU の開発 '14~
 - 農業機械の次世代電子制御ソフトウェア基盤の開発 '14~

進め方

- 初めに **10分**
- モデリングの歴史を振り返る **10分** 休憩
- sysML 2.0 の提案や課題認識をみてる **10分**
- 最近提案されているモデリングを俯瞰してみる **10分** 休憩
- 皆で背景や課題(モデリングの長所、短所、限界)を考えてみる
10分
- これからはこれが良い！と思うモデリングを皆で議論してみる
10分
- まとめ **10分**

この機会だから試してみよう！

- ポジティブシンキング (脳はあなたの思いを支援する方向に動く)
- ポモドーロテクニック (30分集中して、さっと休憩)
- デザイン思考 (思い立ったら、まず行動、作って考える)
- 願いを実現する空海メソッド (①心で思い、②声にだし、③体を動かし顕す)
- トップダウンシンキング (すべて肯定から思考構築を始める、今日はこれ)
- ボトムアップシンキング (すべて否定し、納得した物だけ脳内思考構築)
- タグ付けシンキング (情報に納得度のタグをつけ取り込んでいく、ニューラルネットワーク的思考とも言える Best!)
- 言霊(ことだま)思想 (人工知能/ディープラーニング/脳科学の時代なので、温故知新いろいろ関心を持ってみよう)
- 本をたくさん読みましょう！ Webから情報をインプットしましょう！ わからなくてもタグ付けして頭に溜め込みましょう！ 解らなかつたら何度も読みましょう！
- **頭の中がもやもやすることこそ、エンジニアの成長に必要！** そのためによく理解できていない情報が大量に頭の中に存在することが効果的！
行動、経験がフィットすると “あっ、あのことか！” という感動に！



1 モデリングの歴史を振り返る

- モデリング とは？
- C、Java言語があるじゃないか！
- モデリングにすると、なぜ良いのか？
- モデリングの、そもそもの注意点

モデリングとは？

UML と、Matlab/Simulink、sysML

- 仕様書などシステム記述やMBSEシステムズエンジニアリング目的の sysML、
(Model Based Systems Engineering)
- 制御/シミュレーションに注目したMatlab/Simulink の MBD
(Model Based Development)、
- ソフトウェア設計表現のしやすさに注目した UML の MDD
(Model Driven Development)

別側面

- プログラミングのためのモデリング
- 思考を助けるドローイングのモデリング
- シミュレーションできるモデリング

C、Java言語があるじゃないか！

C言語の長所と欠点。(欠点が多い)

- コントロールフロー表現は得意
- データフロー表現は苦手
- 部品化が苦手
- 階層化が苦手
- アーキテクチャ表現が苦手
- 会話型開発、プロトタイピングが苦手
- 抽象化記述が苦手。
- データの状態属性が苦手 (nil概念 (Liveness)、mutable、参照透過性)。

C言語は40年前 真空管時代に提案された言語
(モデリングは、注意をすれば上記を補える)

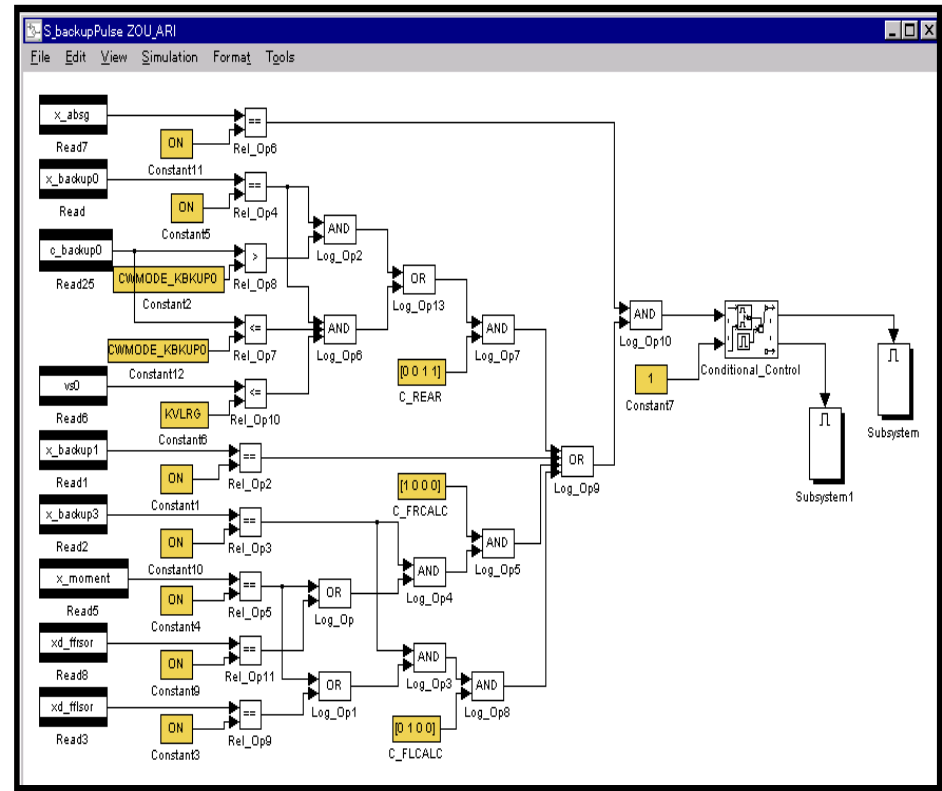
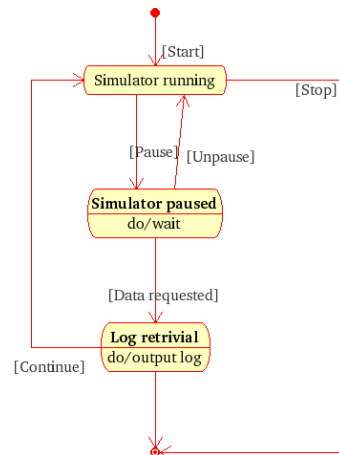
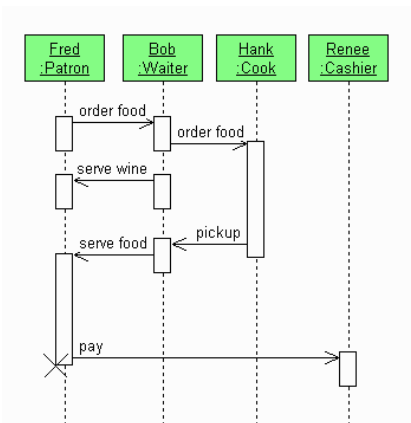
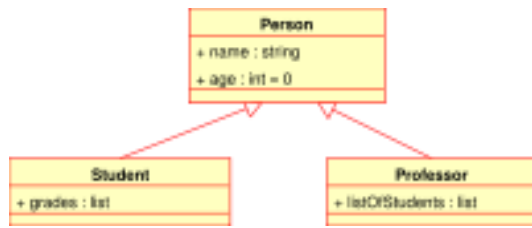
モデリングとは？

UML (MDD) と Matlab/Simulink (MBD)

UML

Simulink

Unified Modeling Language



MDD と MBD

(UML Matlab/Simulink)

表現力は高いが、プログラミングできない、
ドローイングツール

表現力は限られるが、コード
生成、シミュレーションできる

モデル名		MDD	MBD
構造モデル		○	—
分析モデル		○	コントロールフロー相当
シナリオモデル	ユースケース図	○	—
	シーケンス図	○	要求&シミュレーション結果
	コミュニケーション図	○	—
	アクティビティ図	○	データフロー相当
制御対象モデル		—	○
制御装置モデル		—	○

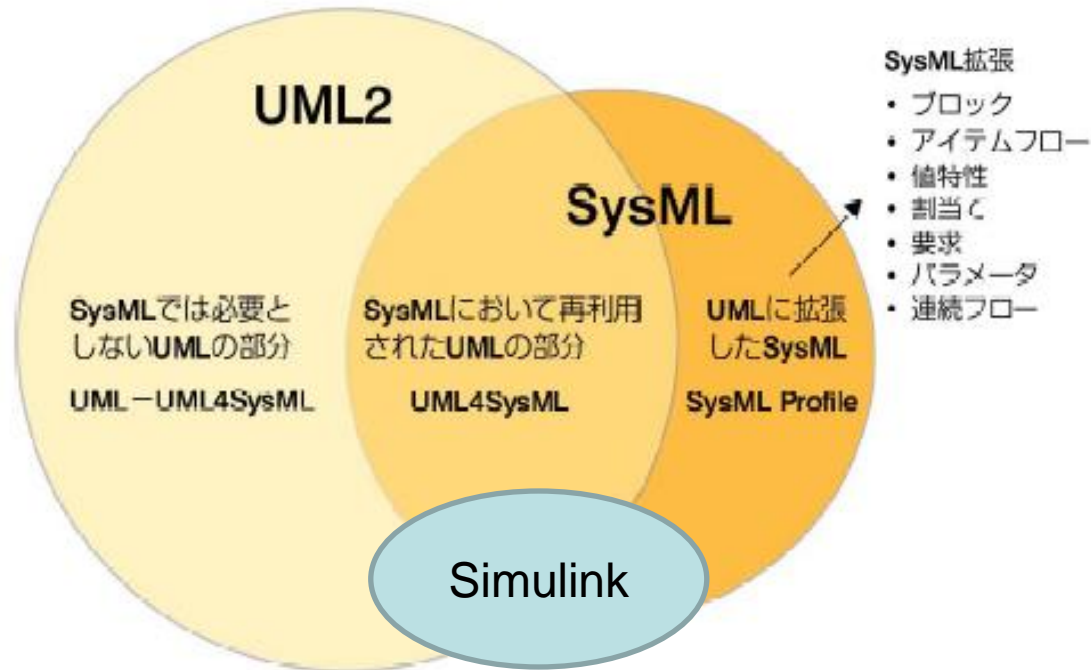
表 2-1 MDD と MBD のモデル比較

(平成23年度モデルベース開発技術部会活動報告書

<http://www.ipa.go.jp/files/000026871.pdf>)

システム仕様記述のsysML (MBSE)と ソフトウェア設計のためのUML (MDD) 制御記述とシミュレーションのためのSimulink (MBD)

■ SysMLとUML



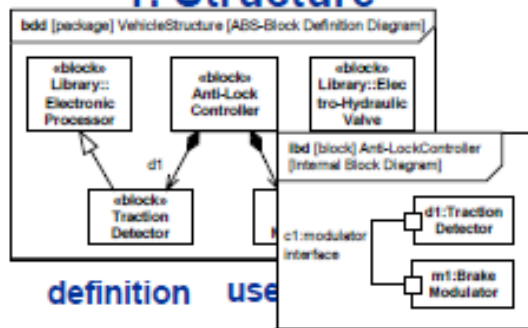
Copyright © 2006-2008 by Object Management Group.

http://www.object-report.jp/archive/SysML_Tutorial_2008.pdf

SysMLの4本柱：ABSの表記例

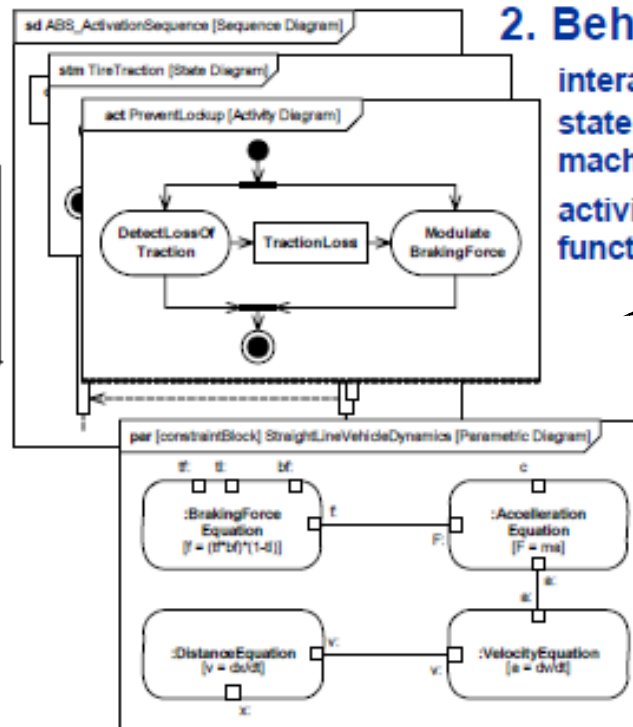
ABS：アンチロックブレーキ制御装置

1. Structure



definition use

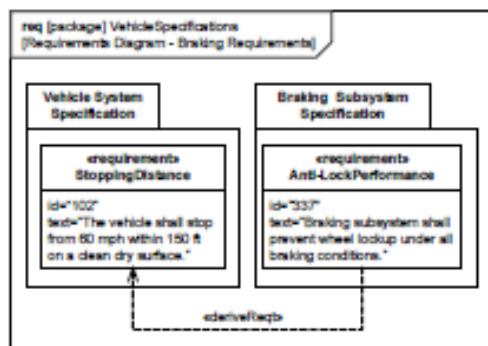
2. Behavior



interaction
state
machine
activity/
function

ヘテロジニアス
ハイブリッド
階層記述
が可能

3. Requirements



4. Parametrics

1. 構造
定義
利用
2. 振舞い
相互作用
状態機械
活動/機能
3. 要求
4. パラ

モデリング比較まとめ (参考)

	ヘテロジニアス	ハイブリッド	アーキテクチャ	振舞い
sysML	○	○	◎	○
Simulink	○	○	○	△
ADLアーキテクチャ記述言語	○	○	◎	×
UML	×	×	○UML2 △UML1	○
Modelica	◎	○	○	△*

*Modelica is a Modeling Language, not a programming language

モデリング とは？ まとめ

- **SYSML、Matlab/Simulink**はヘテロジニアス、ハイブリッドな領域の取り扱いが便利
 - = メカ設計者まで含んだ コミュニケーション可能。
 - Simulinkは実行可能な仕様書が実現できるといわれる。
- **UML**はESLレベルのツール。エレクトロニクス技術者には良いが、**メカを含むヘテロジニアスな領域の**扱いは難しい。(ESL: Electronic System Level)
 - **UML**はハイブリッド制御システム記述も苦手。
 - **UML**もヘロジニアス／ハイブリッドな領域へ進化中 (UML2.0やsysML)
- **sysML**はシステムレベルの仕様書をモデリング記述することを目的に標準化された。

今回は昨年から動き出したsysML2.0の動きから、現在システムモデリングで何が必要と考えられているかを、確認します。

実は仕様書は、なかなか完成していない。
(出来るものではない)、
さらに仕様書だけでは検証できない？

■ 事例:

仕様書通りのソフトウェアであるが、悪路を走ったら、
センサーの誤ダイアグを検出した。

- どのくらい悪路を想定しているか、どこかに記載していなければ、検証できない。(根拠)

UML、sysMLは
10年進化していない
MBD, MDDへの要望も広がっている。

プログラミング記述をさらに抽象化したモデリング (例UML'97標準化)にも欠点があった。

■ 当初UMLは、ただのオブジェクト指向記述モデリング手法(欧州認識:そのためUML2.0へ改良折込したが不十分)

- コントロールフロー表現が苦手(UML2からSDL概念:アクティビティ図追加)
- データフロー表現が苦手(UML2からコンポジットダイアグラム追加)
- 部品化が苦手、デザインパターンのみを部品化手段として提供(UML2より コンポジットダイアグラム追加)
- 階層化が苦手(UML2からコンポジットダイアグラム追加したが不十分)
- アーキテクチャ表現が苦手(UML2からコンポジットダイアグラム追加)
- ADL概念が弱い(UML2からコンポジットダイアグラム追加)
- 抽象化記述しにくい。(UML2からメタモデルインフラを追加)
- メタプログラミング(ドメイン固有な抽象的な記述)が弱い
- プロダクトライン表現が弱い(未対応: EAST-ADLではフィーチャー図を追加)
- ヘテロジニアス、ハイブリッド記述が弱い(未対応: sysMLではパラメトリック図を追加)

■ この理由もあり、MATLAB/Simulinkの応用が拡大している。

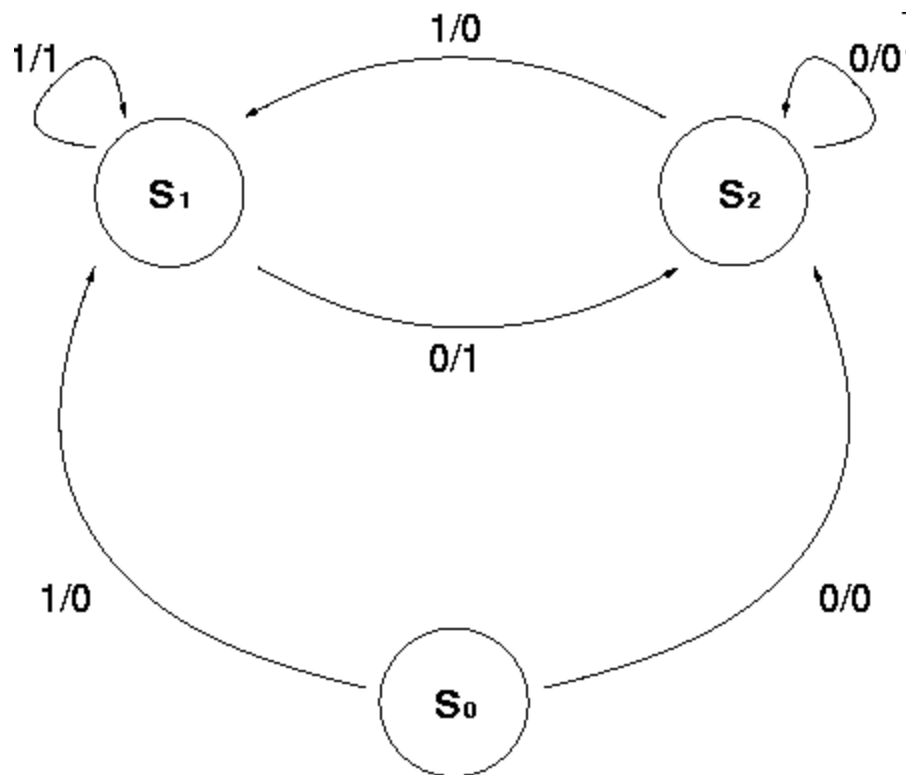
■ DSMモデリングや依存性表現モデリングも必要で拡大。

■ さらにプロダクトラインのフィーチャー図、説明能力側面でGSN、D-CASE、システム記述でsysML

最近の状況

モデリングの、そもそもの注意点

長所、短所認識：モデリング表記の変遷（過去） 状態遷移図(1/2)



ミーリ・マシン (Mealy Machine) : 1955年

ハレルの状態遷移図
(デビッド・ハレル氏が
1987年に開発)

<http://ja.wikipedia.org/wiki/状態遷移図>

モデリング表記の変遷(過去) 状態遷移図(2/2)

認識

状態遷移図は
対象システムが大規模化すると、
数100枚に及び、
そうになると
全体がさっぱりわからない。



改良：表記は“階層化”され“記述能力”強化、大規模化へ対応

- 状態遷移図：SF → Matlab/simulink/statechart, UML2/状態遷移機械図
- 状態遷移表：ZIP-C (キャッツ社) → 階層化状態遷移表(HSTM)

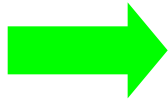
1997年頃

階層化
状態遷移図を
必ず使ってください

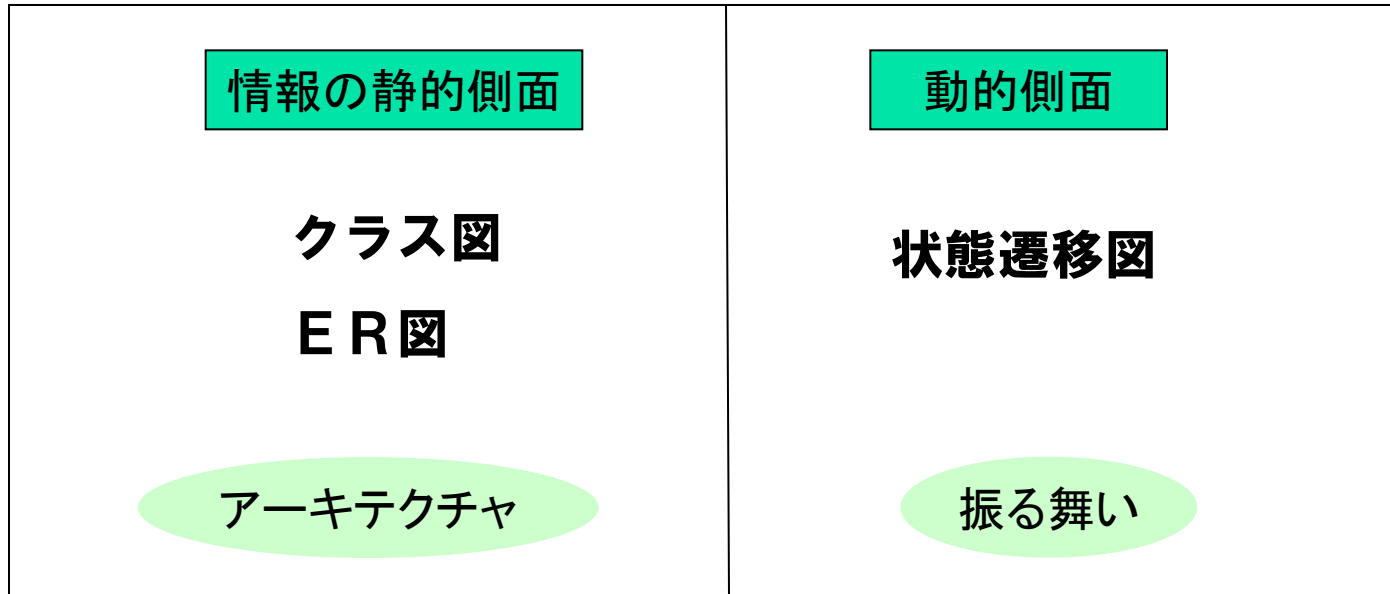
認識

それでも、
さらに大規模化すると、
やはり
全体がさっぱりわからない。

静的構造と動的構造の認識



-“アーキテクチャ”と“振る舞い”に分ければ、判り易いのではないか？-

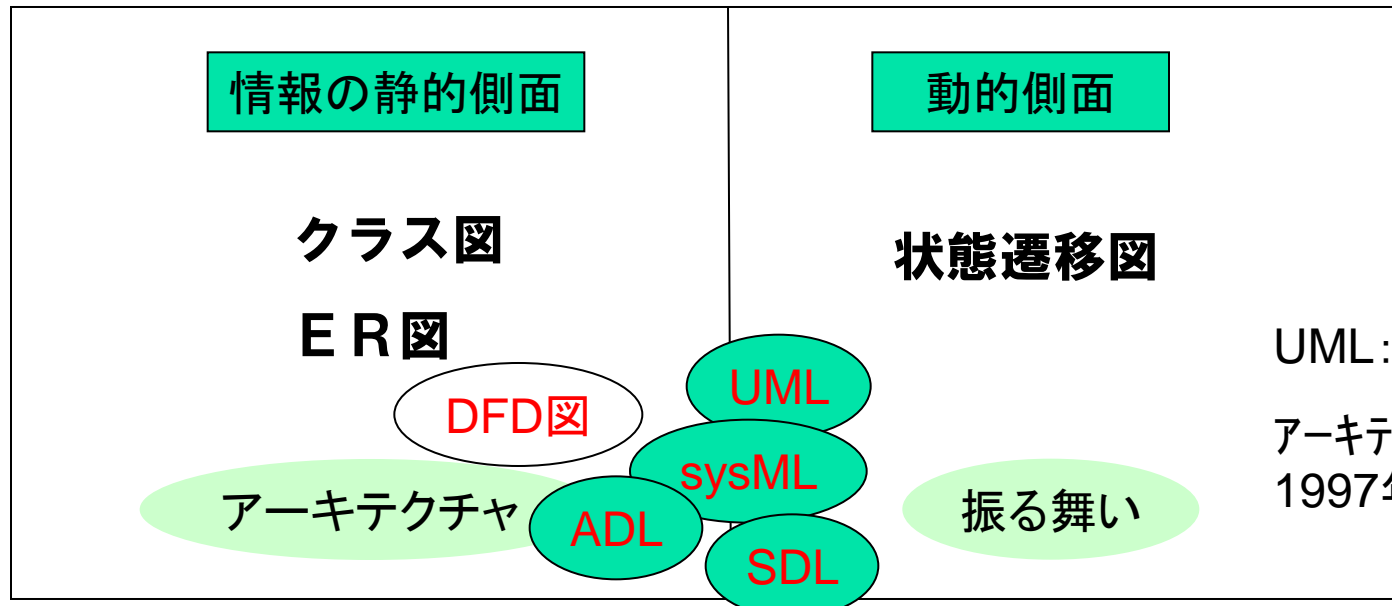


- プログラム = データ構造 + アルゴリズム
- プログラム = アーキテクチャ + 振る舞い

静的構造と動的構造の認識



-“アーキテクチャ”と“振る舞い”に分ければ、判り易くなるのではないかと考えた-



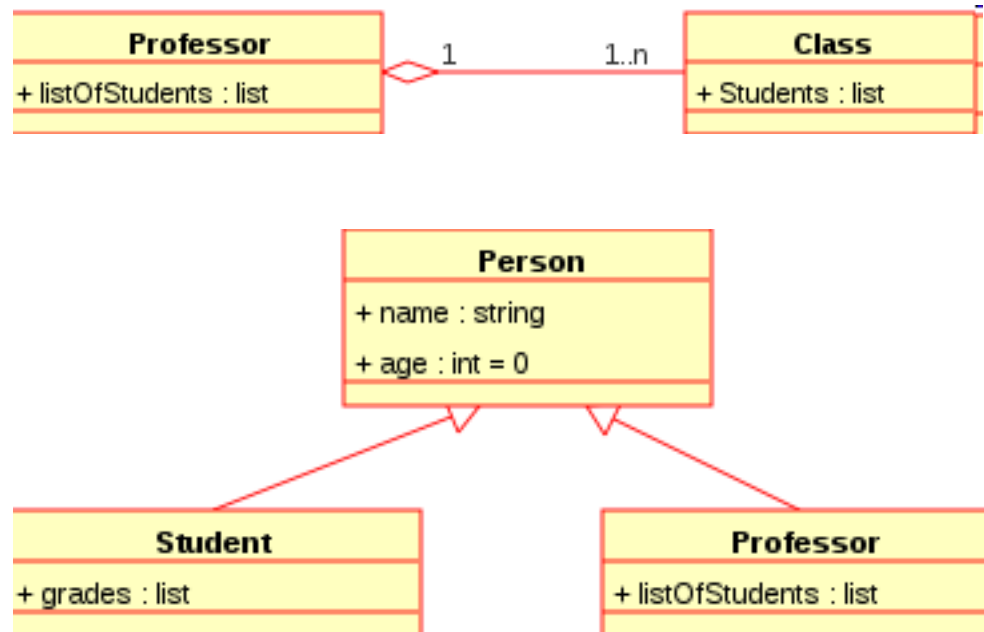
- プログラム = データ構造 + アルゴリズム
- プログラム = アーキテクチャ + 振る舞い

長所、短所認識：モデリング表記の実情

静的構造：クラス図(1/2)

オブジェクト指向表記例(UML)

クラス図：
1995年頃



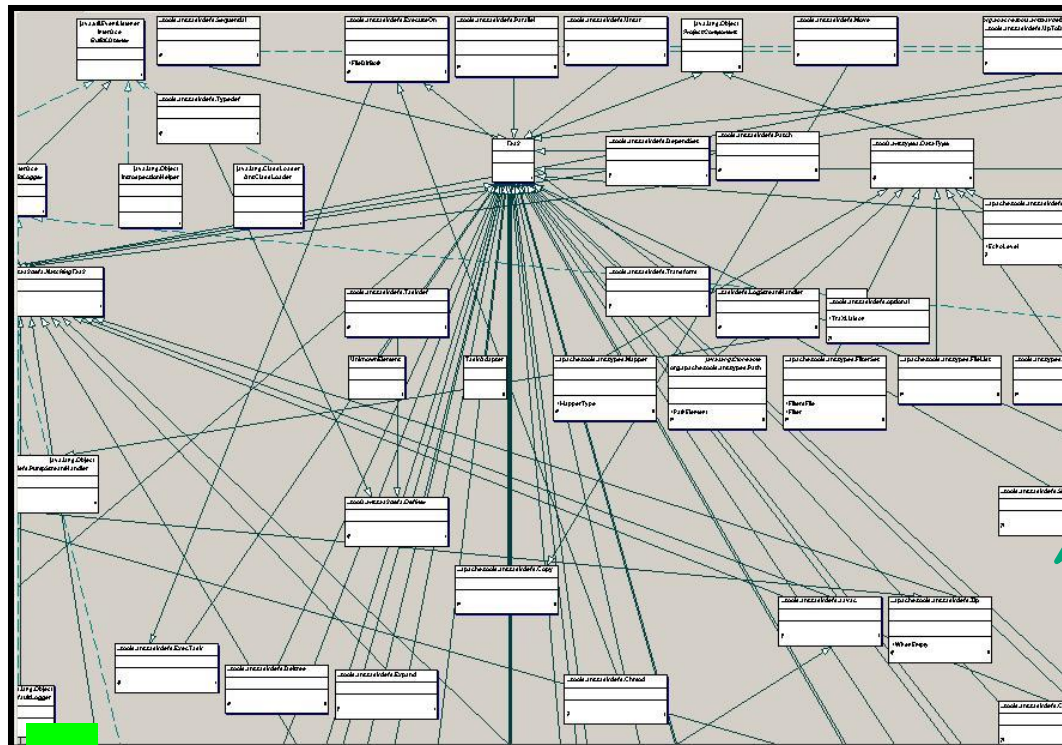
<http://ja.wikipedia.org/wiki/クラス図>

モデリング表記の実情

静的構造：クラス図(2/2)

オブジェクト指向表記例(UML)

欧州では
2001年頃の
認識



クラス図で
大規模システム
を整理できる
という認識では
誤りです

<http://bizmo.co.jp/>

認識

静的構造を表すクラス図も、大規模化すると、
やはりシステム全体がさっぱりわからない。



現在のまとめ：静的構造は多面的に

- 階層化は第1に必要。
- 関心事(多数)への、多種の適当な多面的表現が必要。
- 対象の違い(メカ、ハード、ソフト)の表現方法、あるいはそれらの統一表現が必要。



静的構造(アーキテクチャ)と、
その記述技法、標準化が重要。



システム/アーキテクチャ記述技法

sysMLはクラス図からの階層化に一つの提案をした。

2000

2000年よりオブジェクト指向は反省期へ



Results

資料事例1

Serious UML 1.x weaknesses detected!*

- missing concept hierarchical decomposition
 - divide and conquer (top down)
 - building blocks (bottom up)
 - bidirectional interface
- missing concept for logical components
- missing concept for mapping to physical components
- missing concept for encapsulation
 - black-box approach for just the right level of detail
- ...



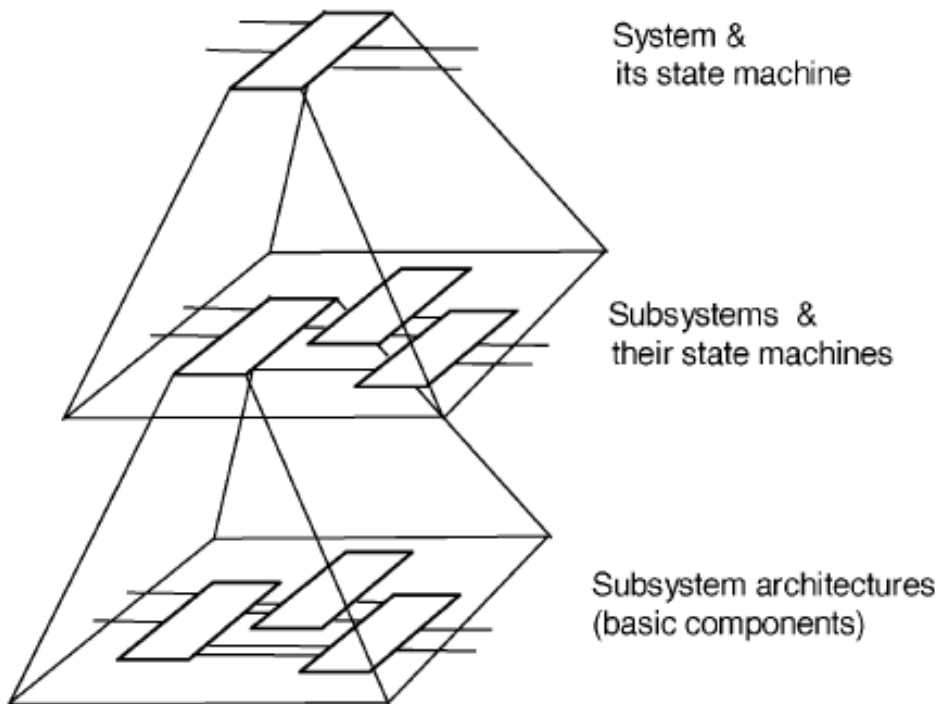
UMLにおける
ADL概念の弱
さを指摘

*M. Broy, M. von der Beeck, P. Braun and M. Rappl: *A fundamental critique of the UML for the specification of embedded systems, unpublished, 2000*

ソフトウェア構造の ブロック的、階層化のあるべきイメージ

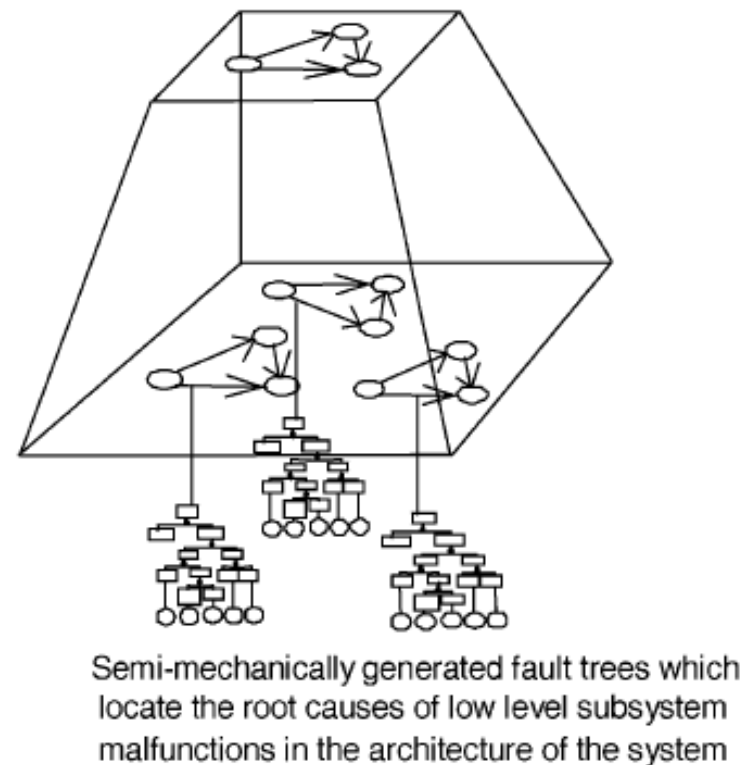
階層構造モデル

Architectural diagrams of the system and its subsystems



振る舞いモデル

State machines of the system and its subsystems



現場エンジニアリング(モデリング編)

UML,sysMLでの記述順、詳細化記述順のポイント

- UMLでは、従前のクラス図中心のオブジェクト指向にプラスし、アーキテクチャ指向を織り込み具体構造化していく記述順が、時代の流れとなっている。(特に組み込み)
- UML1.4では不足で、UML2.Xでは織り込まれてきた(少し織り込まれた、との表現が順当)。さらに思い切った折込がsysMLの位置づけ。
- 具体的には、ある程度設計が構想されたら(ユースケース、クラス、シーケンスができた)構造化ブロック図で詳細化を考えましょう...の順。
- ブロック図は、UMLではコンポジットダイアグラムと呼ばれ、“ブロック図”名では古臭く、オブジェクト指向では重視されなかった経緯があるので、復活は遠慮がちな名前になっている)
- sysMLではストレートに“内部ブロック図”という名前になっている(その点 sysMLは遠慮なくオブジェクト指向偏重を切り捨てている)。
- ブロック図はある意味データフロー図。
アクティビティ図はデータの流れを左から右へ、制御の流れを上から下へ、同時に書ける、注目される図(それが意図された)。従って、ブロック図から、制御をプラスした構造へ、詳細化できる。
- ただ、あまりに自由に書けてしまうので注意。スイムレーンやブロック図整合を意識する。スイムレーンは、シーケンスチャートから想像できる。
- また、シーケンスチャートから状態遷移図は、本来変換できる。(UML2からシーケンスチャートに分岐を記述できるようになり、状態遷移図整合が、布石で織り込まれた)ので状態遷移図で整理し、アクティビティで詳細化を、考える。(必要に応じて)

‘モデル駆動型開発MDD、MBDさらに MBSEの流れで、現場で意識が必要な点(まとめ)

- **モデリングにすると、なぜ良いのか？**
- C言語の弱点を、アーキテクチャのわかりやすいモデリングで包み込もうとしているのが、MBD、MDD。
- さらにシステム要求から連続させる、MBSE。
- **モデリングの、そもそもの注意点**
- 大規模システムでは**階層化表記**が出来なければ、破綻します。また“アーキテクチャモデル”と“振る舞いモデル”を併用しないと、やはり破綻します。過度の状態遷移図依存は注意ください。
- システムズエンジニアリングでは、ヘテロジニアス、ハイブリッド表記が必要です。

UML,sysMLモデリング規格では具体記述のない、 現場エンジニアリング

問題提起

- 状態遷移図で完全であっても、状態遷移表で確認すると、ユーザ目線では漏れがあります。状態遷移表で確認をしましょう。
- 状態遷移表で完全であっても、安全分析をすると、システム目線では漏れがあります。多種の安全分析をしましょう。システムを意識しましょう。
(これは、何々分析の事例です、他にも多数ある分析作業こそが設計作業です)
- 派生開発、車種展開ではプロダクトライン開発を意識し、フィーチャー図で確認をしましょう(後述)。
- これらは、カスタマー目線の何々分析に関係する基本課題です。
- システムズエンジニアリングでは、この作業は誰かが必ず確認しているはずで、誰でしょうか？
- モデリングsysMLは上流要件を記載する場ですが、結果を記載する場であって、上記を推進する道具ではありません。良いテンプレートが必要です。さらに要件や設計が妥当である説明が求められてきて、わかりやすく、説明するためのモデル表記(GSN、D-CASE)を併用する方向になってきています。いずれにしろ、ゴール(目的)を意識しましょう。

‘最近のMBSE, sysMLを取り巻く周辺の現場関心事、動向(MBD,MDDを含む)

- (参考)基本であるモデリング標準化の停滞、認識遅れ
- ソフトウェア設計用のUMLと、システムズエンジニアリング&仕様書作成用のsysMLと言われますが、UML2.0で織りこめきれなかった改良点をsysMLでは先進提案的に織り込まれている、という位置づけと認識しています。
- 私はソフトウェア設計もsysMLで行ったほうが良いという立場です。

(参考)モデリング標準化の停滞

問題提起

- UML1.0(^97)、2.0大幅変更(^04)その後大幅変更無し
 - ローエンドMARTE、fUML、EAST-ADLの動きは活発だった。
- sysML1.0(^04)その後大幅変更無し
- UML2.0、SysML1.0の功労者は両方ともCris Kobryn氏



■ xUML.org vs U2Partner.org /SysML.org
ステファン・J・メラール氏
https://en.wikipedia.org/wiki/Stephen_J._Mellor



Cris Kobryn氏 <http://kobryn.com/>

- **Cris Kobryn氏**の意思を引き継ぐ人が現れていない
- **日本が次世代をリードもありえる！**
- **吉報！？ 2015年夏 欧州がsysML2へ動き出した！**

(参考)モデリング標準化の停滞

問題提起

事例:

- プロダクトラインのフィーチャー図を織り込む作業が進んでいない
- 状態遷移表はなかなか織り込まれない
- DSM提案(状態遷移表同様表的表現)も効果的なモデリング
- Matlab/Simulinkスタイルガイドは進んでいるが、sysMLスタイルガイドが標準化に至っていない(safeML提案あり)
- MBSEを補完するGSN、D-Case、SCDL(SCN)などの提案が始まって、融合が歩みだした。
- MBSEもMBD/MDDも安全分析など“何々分析”をどう支援／織り込むかは途上 (STAMP、安全分析、脅威分析、何々分析. . .)

2 sysML 2.0 の提案や課題認識をみてみましょう

欧州の課題認識がわかります！（あえて英語のままです、意外とわかります）

- Model Formalism Concept: SysML v2 to be computer interpretable.
- Model Analysis Concept: logical analysis and quantitative analysis in the concept.
 - the quantitative analysis can impose to integrating with other engineering analysis models and tools.
- Model Construction Concept: Consider the 3 model construction use cases
 - from scratch
 - Reusing an existing model
 - Importing unstructured legacy data (e.g. word, excel)
- Model Visualization Concept: concise description of the model view controller (MVC):
- Model Management Concept: clarify versions versus revisions.
- PLM/MBSE Integration Concept: PLM/MBSE and model interoperability.
- MBSE Collaboration & Workflow Concept: Model Interoperability and Standardized API:
- Model Interoperability and Standardized API: Include https in the web standards, synchronization in a collaborative environment
- Systems Engineering Concept Model (SECM): views of different abstraction levels
- Systems Engineering Concept Model (SECM) Property & Expressions Concept: the need for dimensional analysis

Reston Meeting - March 2016 - SysML V2 Planning and Requirements Meeting

Meeting Overview

The following summarizes the objectives and agenda for the System Modeling Assessment & Roadmap WG in Reston, Virginia on March 15 and 17, 2016.

Meeting objective:

1. Review Proposed Systems Modeling Environment (SME) Concept
2. Discuss preliminary approach for SysML v2 RFP (pending satisfactory maturity of the SME Concept)

Meeting agenda:

Tuesday, March 15, 2016

- 09:00 - 09:30 Introduction - Sandy Friedenthal
- 09:30 - 10:30 Modeling Formalism - Yves Bernard (*refer to presentation below*)
- 10:30 - 11:00 Break
- 11:00 - 12:00 Analysis Concept - Manas Bajaj (*refer to presentation below*)
- 12:00 - 13:00 Lunch
- 13:00 - 14:00 [Model Construction](#) - Ron Williamson
- 14:00 - 15:00 [Model Visualization](#) - Chris Schreiber, Josh Feingold, Marc Sarrel
- 15:00 - 15:30 Break
- 15:30 - 16:00 - Laura Hart [Model Management](#)
- 16:00 - 16:30 Integration with PLM - U Kaufmann, M Pfenning (*refer to presentation below*)
- 16:30 - 17:30 Workflow and Collaboration - Hedley Apperly (*refer to presentation below*)

Thursday, March 17, 2016

- 09:00 - 09:15 Introduction & Recap
- 09:15 - 10:15 [Systems Engineering Concept Model \(SECM\)](#) - John Watson (*refer to presentation below*)
- 10:15 - 10:45 Break
- 10:45 - 12:00 [Property Model](#) - Hans Peter de Koning
- 12:00 - 13:00 Lunch
- 13:00 - 14:30 [Model Interoperability and Standard API](#) - Axel Reichwein
- 13:00 - 14:30
- 14:30 - 15:00 Break
- 15:00 - 16:30 SysML v2 RFP Approach and Scope
- 16:30 - 17:00 Next Steps - All

Each Concept Lead presents their concept including the following:

http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_mar_2016_meeting_presentations

sysML 2.0へ動き出している状況

OMG: System Modeling Assessment & Roadmap Working Group Meetings

SysML v2-Meeting

- Cambridge Sep 2015 Meeting Presentations
- La Jolla Dec 2015 Meeting Presentations
- Reston Mar 2016 Meeting Presentations
- Orlando June 2016 Meeting Presentations

http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:reston_mar_2016_meeting_presentations



sysML 2.0へ動き出している状況

OMG:

- [00-SysML v2-Meeting Summary-Reston-friedenthal-2016-03-18-a](#)
- [SysML v2-Modeling Formalism-Yves Bernard-Reston-2016-03-16](#)
- [SysML v2-Analysis Concept-Reston-Manas Bajaj-Intercax-2016-03-16](#)
- [SysML v2-Colaboration & Workflow Concept-Reston-Hedley Apperly-2016-03-16_v4](#)
- [SysML v2-PLM-MBSE Integration Discussion-Reston-Uwe Kaufmann-2016-03-15](#)
- [SECM Status, OMG Tech Meeting, Reston VA, Systems Modeling Roadmap and Assessment Mtg 03-17-2016](#)



欧州は、現状で機能安全、sysMLなどで何が不足と考えているか？

- IoT、自動走行などエリアに参入する台数が予想できない(open、adaptive)
- 接続される各種機器でアップデートがされるようになる (adaptive)
- 公道での実証実験は、不具合が起きれば市民を巻き込む事故に発展するリスクが高く、実験ごとの検証の正しさ、アップデートの正しさを短期間に検証判断するリスクがどんどん高くなる、企業の命運を左右する。どのようにOKを判断すればよいのか悩む
- sysMLを厳密にして(可能なら)検証、シミュレーションが動くようにして、Design by Contract (assertion)で、インCREMENTAL、リアルタイムに(短期に)検証できる仕組みを構築したい。

関連する

最近の欧州FP7組み込み関連大規模技術先行プロジェクトの動向

- ・欧州で進行している、日本では例えばSIP/FIT相当の100社/100億円助成で動いているプロジェクトが複数活動しています。
- そこでは横断的基盤や将来の標準化やシステムズエンジニアリングを目指した取り組みが行われています。
- ・自動走行、IoT、ロボット、CPS、SoSに向けた機能安全 IEC61508rev3、ISO26262rev3改定(2021DIS予想)への先行的活動や関心事例を説明します。
- ・それらと連動して、システムエンジニアリングモデリングsysMLの時代に合わせた2.0への久々の改定の動きや、Modelica、MatlabとFMI+リモートIF(OSLC)で連動したリアルタイムシミュレーションメカニズム構想などの活動事例を説明します。

最近の欧州FP7組み込み関連大規模技術 先行プロジェクトの動向

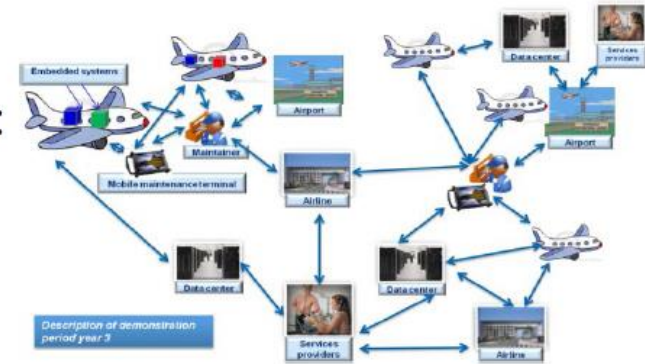
- **ARROWHEAD: Service Interoperability Enabling Collaborative Automation for Production and Energy System Automation, Intelligent-Built Environment and Urban Infrastructure**
- **EMC²: Platform for Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments**



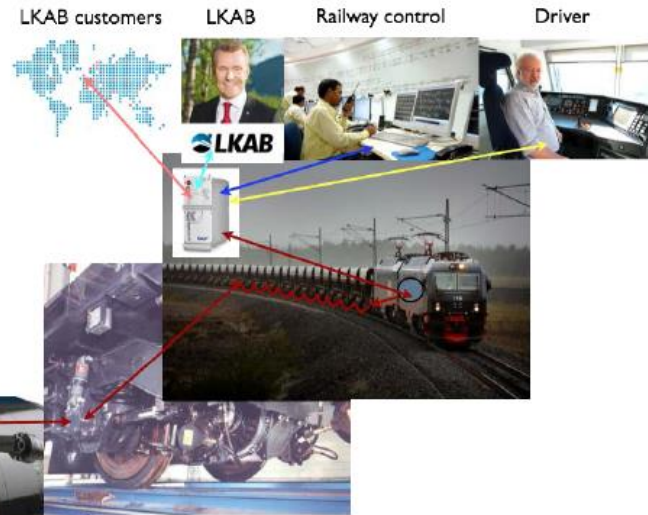
Project ARROWHEAD

To be demonstrated in real world applications:

- The virtual market of energy
- Production efficiency e.g. manufacturing, mining, airlines, automotive, water, ...
- Energy system optimization
- Smart cities
- Electro-mobility
- From device to enterprise management in the collaborative automation cloud



Stakeholders





Project Overview

Embedded Multi-core Systems for Mixed-Criticality Applications in Dynamic and Changeable Real-Time Environments – EMC²

(Artemis Innovation Pilot Project (AIPP))

- AIPP 5: Computing Platforms for Embedded Systems
- Coordination: Infineon Technologies AG
- Consortium Size: 99 Partners, 16 EU Countries + Israel
- Many challenges:
 - evolving adaptive Systems-of-Systems,
 - Safety & Security by Design and during Life Time (run-time assurance),
 - multi-core, mixed criticality

not resolvable at current State-of-the-Art in methodology, standards and approaches

- → new approaches in IEC 61508-3, preparation of Ed. 3.0.

活動を説明する資料

- Adaptive Dynamic Systems (SoS) Holistic Safety & Security Assurance – New Approaches and Standardization Activities –

http://www.cpsos.eu/wp-content/uploads/2015/10/CPSoS-WS_Adaptive-Dynamic-Systems_E-Schoitsch.pdf

- Safety Certification of Software-intensive Systems with Reusable Components

<http://www.safecer.eu/images/RigaSeptember2013/presentations/pSafeCer%20FOW%20Influencing%20Practice.pdf>

3 最近提案されているモデリングを俯瞰してみましょう

キーワードでGoogle
検索してみてください

- システムグラム:
- GSN: ゴールストラクチャー
- D-CASE: デペンダビリティケース
- SCDL: 機能安全セーフティゴール記述
- SysML Rev2: 次項参照
- Synchart: 並列記述ステートチャート

最近のMBSE, sysMLを取り巻く周辺の 現場関心事、動向(MBD,MDDを含む)

- 現代先進安全設計の自動車機能安全規格の解釈はJasPar,JARI活動で共有されてきたが、そこから踏み込んだ相場観は、すべては議論できてはない。そのため、ある程度相場観を支援できないか、提起が始まっている。

■ **GSN, D-Case, SafeML, SCDL ,Systemigrams**

GSN: Goal Structure Notation (2011)

システムが達成すべき目的や性質について、その達成を導く方法・思考を可視化する際に用いる記法。MBSEは目的、ゴールを意識する事が大切です。

D-Case: Dependability Case (2013) Assurance Caseとも言う)

GSNを発展させた合意の表記法(一般社団法人 ディペンダビリティ技術推進協会: IEC 62853/62741,IEC60300-1,ISO/IEC15026)

SafeML: Safety Modeling Language

産総研:安全関連情報をコミュニケーションよく交換するモデリング記法、テンプレート

SCDL: Safety Concept Description Language(2013~)

エレメントの表記法、安全要求の表記法、それらを統合した安全コンセプトの表記法(安全コンセプト記法研究会)

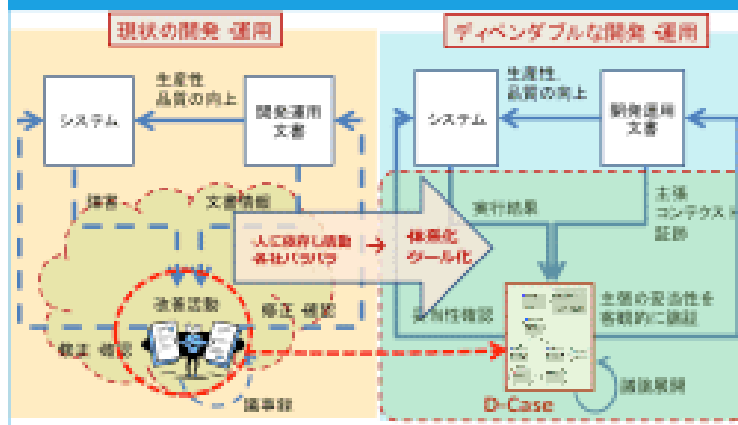
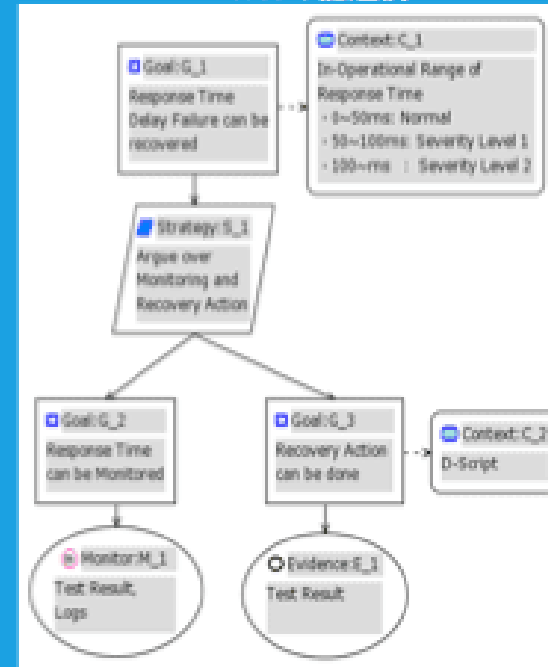
Systemigrams(システミグラム):システムと問題の構造を分かりやすく表現

DCASEの基本であるGSNは非常に簡易な基本表記
 (4種の構造を結ぶ: 2011年イギリスで提案)であり、マインドマップの様に
 一般化すれば、白板を前にした討議でも利用可能

D-Case

- GSN (Goal Structuring Notation) をベースとした合意形成のための表記法
- GSNを拡張し、システムの運用状況の監視と詳細な記録を実現
- 開発・運用を含む全ライフサイクルでのステークホルダー間の合意形成
- D-Case記述の合意に基づく変更履歴が説明責任遂行を支援

D-Caseの記述例



- D-Caseの活用により、属人性の強い改善活動から、体系化・ツール化されたプロセスへ



関連活動が行われている場の事例

- DEOS協会
- INCOSE、INCOSE-J
- SERA研究会
- ASIF 車載組み込みシステムフォーラム
- 名古屋大学 ASDOQ研究会
- 安全コンセプト記法研究会
- 認証工学研究会



4 皆さんと、自分の業務領域を意識しながら、背景や課題(モデリングの長所、短所、限界)を考えてみます

事例:

- 何々分析を支援する機能が弱くないか？
(標準モデルは結果を記入する場)
 - 機能性能分析、安全分析、脅威分析、プロダクトライン分析、ユーザビリティ分析、状態遷移表網羅性分析、などシステムズエンジニアリングの分析は多種
- 昨今話題の並列機能記述は？
 - 並列実行を考えると同期の問題が出てきます
- 昨今話題のゴール思考を明瞭にする必要は？
 - 設計書に、ノウハウや背景などすべてが織り込まれていますか？



5 これからは これが良い！と思うモデリングを 皆さんで議論してみたいと思います

- 用途は？工程をどこで使うか想定してみてください
- 上流？まとめるため？分析の道具？
- 設計のアイデアをわかりやすく、さっとまとめたい？
- 具体的にシミュレーションしたり、コード生成したい？
- 課題抽出をしたい？



演習

課題を思い浮かべ、
それに対する直感を大切にしましょう！

(書き出すことや、言葉に出すことも大切です)



まとめ

- グラフィカルモデリングの実態を共有しました。
- 形成の歴史、課題背景、用途、限界、今後の動向を理解すると、ツールをどう組み合わせるべきか見逃しが無くなります。
- モデリングは、広範に用いられています、どんどん主流になっています。
- 用途を意識しましょう。
- 完璧なツールは、やはり、ありません注意は必要です。
- こんなモデリングが良い！とこれからも提案してみてください！

組み込みソフトウェアの スペシャリスト

The Professional Organization for Software Developments

suzumura-nobuyasu@aisin-comcruise.com 鈴木延保