

これからのモデリング

細合 晋太郎

Change Vision

スケジュール: 10:30～11:40

✳️10:30～11:00: モデリング業界動向

✳️11:00～11:20: 各種ツールデモ

✳️11:20～11:40: Q&A、ディスカッション

Modelingとは

✧抽象、取捨

- ✧膨大な情報から欲しい情報のみを抽出する
- ✧要らない情報を捨てる

✧観点

- ✧いろいろな側面から眺める
- ✧抜け・漏れを防ぐ

✧関連

- ✧モデル同士のつながりを作る
- ✧繋がりから新たな情報を得る

✧分析

- ✧シミュレーション
- ✧モデルを検証する

✧出力

- ✧ソースコード
- ✧ドキュメント

astah*

The screenshot displays the Astah IDE interface with a Java class diagram. The left sidebar shows a project tree with packages like 'actions', 'exception', and 'generator'. The main workspace shows a detailed class diagram with the following components:

- actions package:** Contains `GenerateAction` and `SettingAction`, both with a `+ run(window : IWindow) : Object` method.
- setting package:** Contains `SettingManager`, `GenerateSetting`, `SettingDialog`, `HashMap<String, GenerateSetting>`, `JComboBox<TemplateEngine>`, `JComboBox<String>`, `Map<String, GenerateSetting>`, `TemplateEngine`, `TemplateMap`, `TemplateType`, and `JComboBox<TemplateType>`.
- xtendhelper package:** Contains `Function1<T,R>`, `Using`, `Consumer`, and `Consumer<String>`.
- exception package:** Contains `BuildException`.
- test package:** Contains `SettingDialogTest`.

Key relationships in the diagram include:

- `SettingManager` inherits from `HashMap<String, GenerateSetting>`.
- `SettingManager` has an association with `GenerateSetting` (labeled `- currentSetting`).
- `SettingDialog` has associations with `GenerateSetting` (labeled `- combo_templateEngine`), `JComboBox<TemplateEngine>` (labeled `* combo_templateSet`), and `JComboBox<String>`.
- `GenerateSetting` has associations with `TemplateEngine` (labeled `- templateEngine`), `TemplateMap` (labeled `* - mapping`), and `TemplateType` (labeled `- templateType`).
- `TemplateMap` has an association with `TemplateType` (labeled `{unique}`).
- `TextBinding` has an association with `Consumer<String>` (labeled `- setterFunction`).
- `Consumer<String>` has a dependency on `Consumer` (labeled `<bind><T>-String`).

いままでのモデリング

✧単一システムのソフトウェア設計

✧UML / SysMLなど標準仕様のモデリング

これからのモデリング

- ✧ソフトウェア屋のためモデルから、関わる全ての人へのモデルへ
- ✧関わる全ての情報の統合と必要な情報の抽出
- ✧ステークホルダーに応じたモデリング、ビューポイント

Modeling業界動向

- ✧ Eclipse
- ✧ Jet Brains
- ✧ No Magic
- ✧ その他

<https://www.eclipse.org/>

Eclipse

Eclipse

- ✧プログラミングのためのIDEから、モデリング基盤へ
- ✧Eclipse Modeling Framework: MOF実装
- ✧e4 : IDEのUI周りをモデルで管理
- ✧Orion: クラウドIDE
- ✧LPM: IDE機能のAPI化

Eclipse CDT 界限

✧ Linux 環境

✧ Arduino 環境

✧ Eclipse をベースにした組込み向け IDE

✧ True Studio 等

✧ Docker でクロス環境構築

<https://www.jetbrains.com/>

Jet Brains

Jet Brains

✧ IntelliJ

✧ Android StudioもJet Brains

✧ Kotlin

✧ その他、各種言語向けIDE (Python, C#, Ruby, PHP, JavaScript)

✧ Jet BrainsのDSL環境

- ✧ Textual DSLだけでなく、Graphicalや表形式、数式の埋め込みなどなど、かなり自由なDSLが定義できる
- ✧ 異なる表現のDSL・コード間もモデルで結ばれているので、変更したら即時に他方にも反映される

✧MPSをベースにした、組込み統合環境

✧DSLによる要求記述

✧ステートマシンDSL

✧C99

✧テスト

✧検証

✧デバッグ

✧これらがDSLですべて繋がる

<https://www.nomagic.com/>

No Magic

Magic Draw

<https://www.nomagic.com/products/magicdraw>

- ✧ SysML2.0、MBSE (Model Base Systems Engineering) の最有カツール
- ✧ 各種モデルの統合や、ステークホルダに応じた ViewPointの定義からViewの生成など、メタ定義とモデル定義をシームレスに行き来できる
- ✧ その他のツールチェーンも強力で、各種ツールとの連携も多い
- ✧ MBSEでの利用事例もいくつか出てきている

その他

✧標準記法の定義から、MBSE (Model Base Systems Engineering) の実現基盤に

- ✧[SysML v2 RFP Working Group](#)
- ✧[OSLC4MBSE Working Group](#)
- ✧[Auto-View Generation Working Group](#)
- ✧[SysML and Modelica Integration](#)
- ✧[SysML Extension for Dynamic Simulators](#)
- ✧[Quantities, Units, Dimensions, Values \(QUDV\)](#)
- ✧[Mapping between SysML and AP233](#)

✧ MBSEのOSS基盤。INCOSEメンバーによる実装？

✧ mms (Model Management System)

✧ モデルリポジトリ。ElasticSearch + PostgreSQL

✧ REST APIでモデルにアクセス

✧ EMS-WebApp

✧ ビュー定義とビューアのウェブアプリ

✧ mdk

✧ Magic Draw 連携。現状Magic Drawからモデル定義を持って
てくるのが一番扱いやすそう

Change Vision 動向

Change Visionの取り組み

✧研究開発中

- ✧EMFを用いたモデル基盤

- ✧独自モデルを作りやすいプラットフォーム

- ✧モデルリポジトリ連携

✧STAMPツール(開発中)

デモ(予定)

✧MPS / mbeddr

✧Eclipse Orion

✧XText

✧astah小技集

Q&A、ディスカッション