

astah上の GDMAツールの使い方

2017/12/11

海谷 治彦

目次

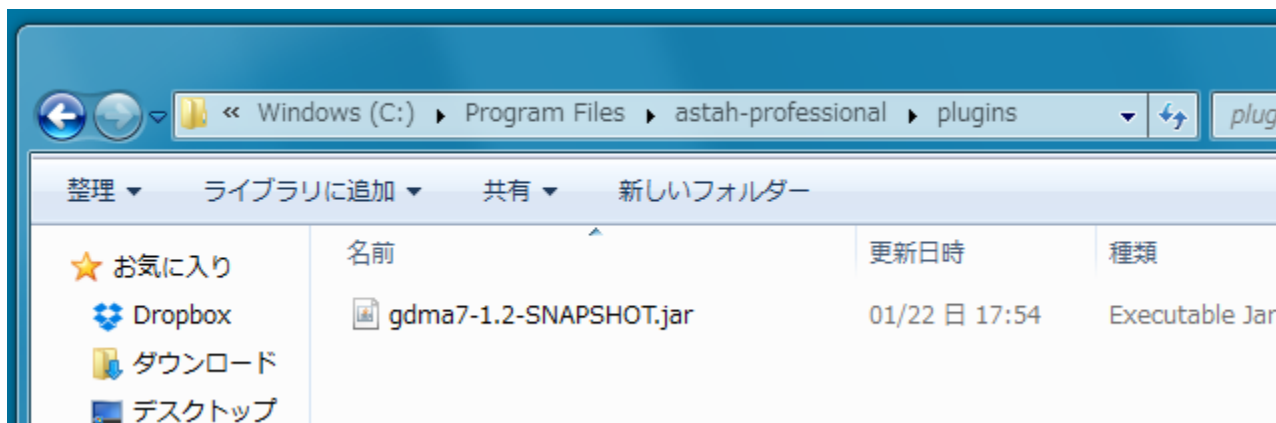
- 主な特徴
- インストール
- 主な機能の説明
- 良さの指標 (metrics)
- 基本的な利用手順
- TIPS

主な特徴

- 前述のSDモデルの記述を少し楽にするツールです.
- 主な機能は,
 1. SDモデルの構文上の誤りを指摘
 2. モデル化した業務の「良さ」を表す指標の提示
 3. 多少の入力支援

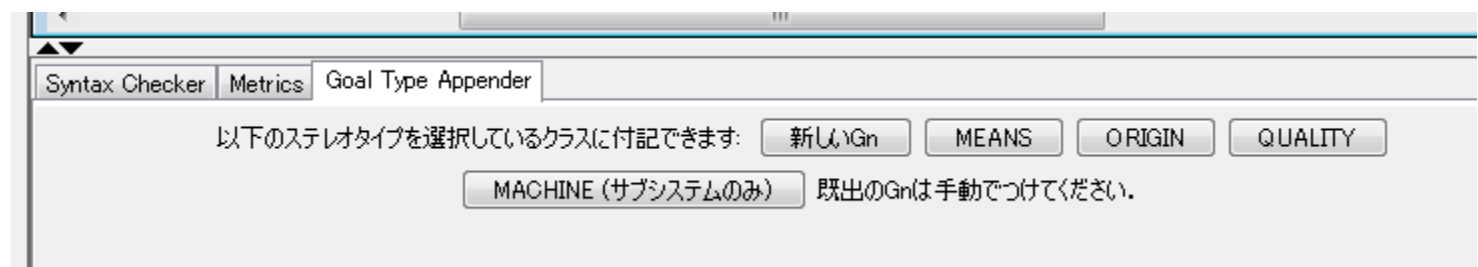
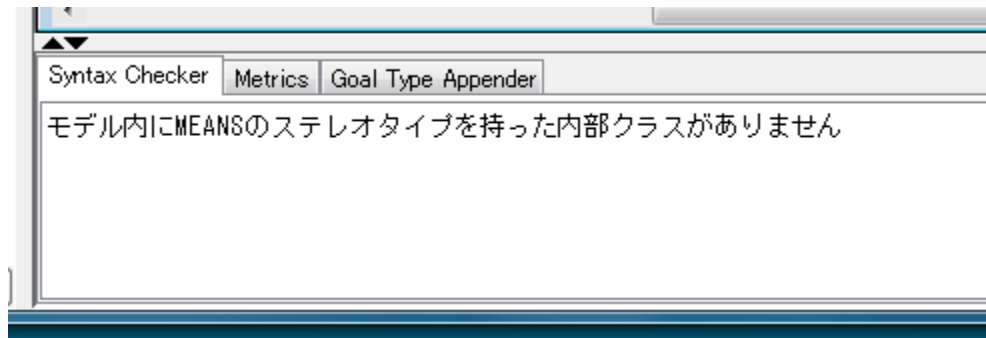
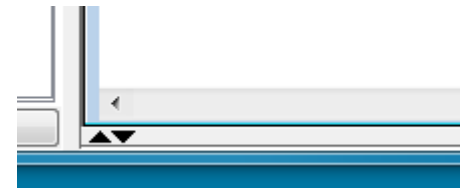
インストール

1. gdma7-1.2-SNAPSHOT.jar をdotcampusからダウンロード
2. このファイルを,
C:¥Program Files¥astah-professional¥plugins¥
にコピー (管理者権限が必要)
3. 以降, 単に astah を起動する.



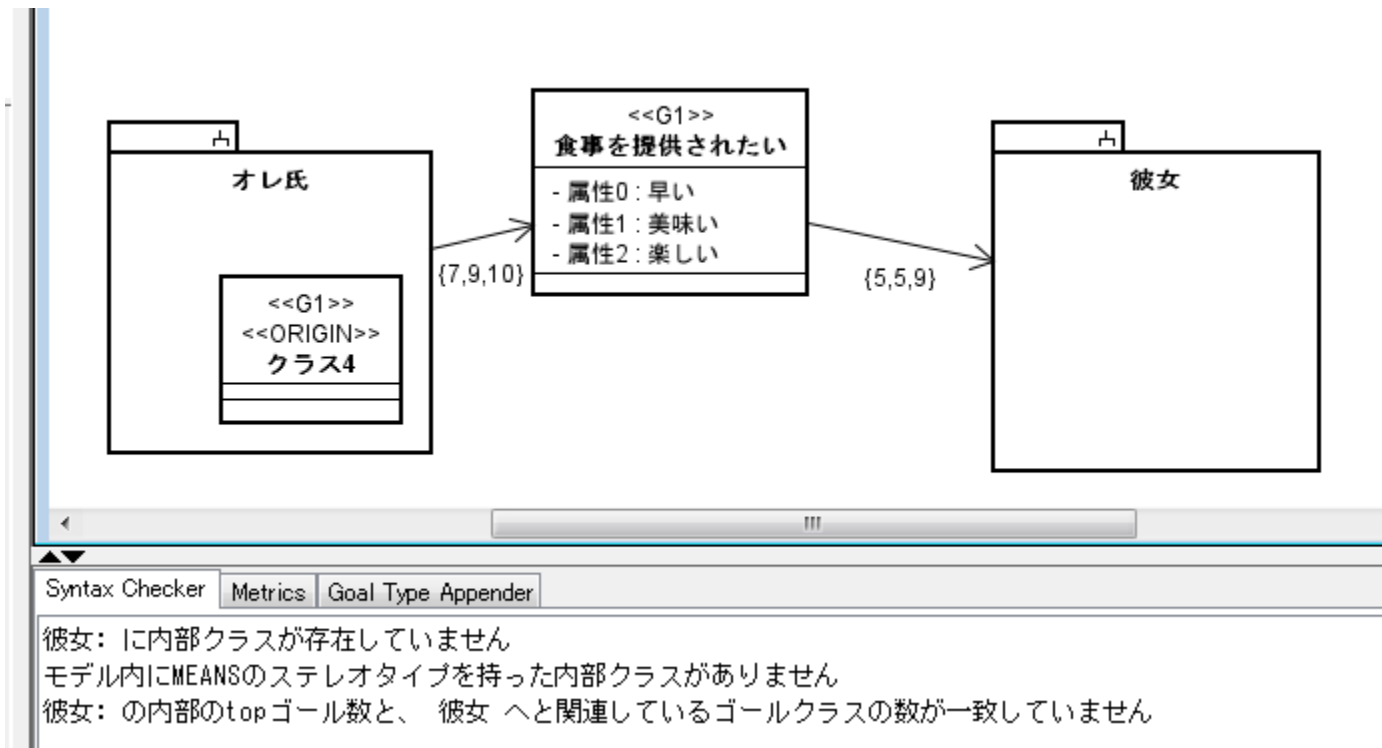
起動確認

- astah を起動して、適当にクラス図を作ると、以下のような3つのタブが出てればOK
- クラス図以外では使わないで下さい。
- タブが出てない場合には、▲を押して。



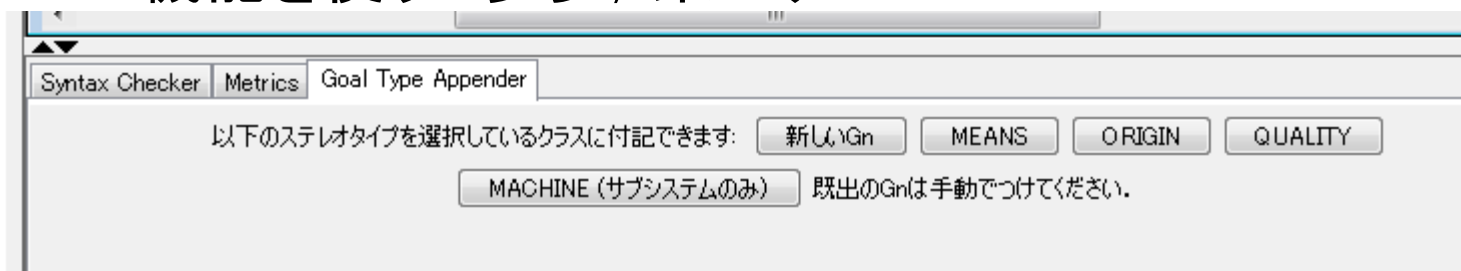
Syntax Checker

- クラス図で書いたSDモデルが文法的に間違っていると警告がでます。
- 警告が全て消えるようにモデルを書いてください。



Goal Type Appender

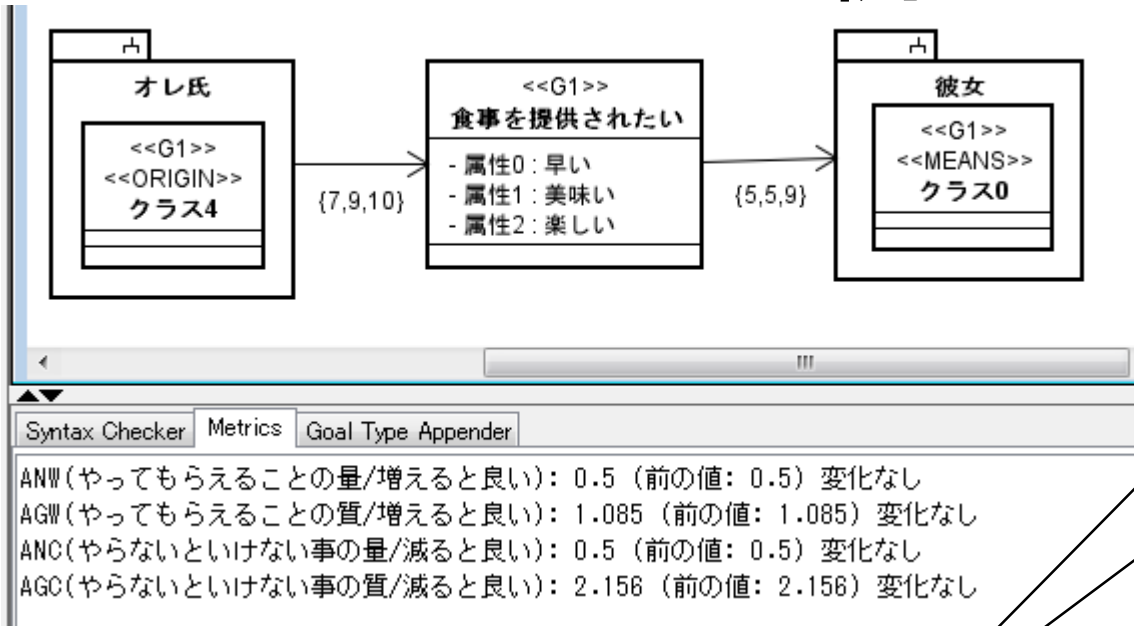
- SRモデル等に特化したステレオタイプをつける補助をしてくれます。
- まず、クラスかサブシステムを1個選択します。
- 以下に対応するボタンを押すとステレオタイプが選択した要素につきます。
 - 新しいGn: 例えば, G5 までは既にあるれば, G6がつきます。
 - MEANS: MEANS が単純につきます。
 - ORIGIN: 同上, ただしクラスのみ有効。
 - QUALITY: 同上
 - MACHINE: 同上, ただしサブシステムのみ有効。
- 別にこの機能が無くてもステレオタイプを付記できますが, この機能を使うと多少, 楽です..



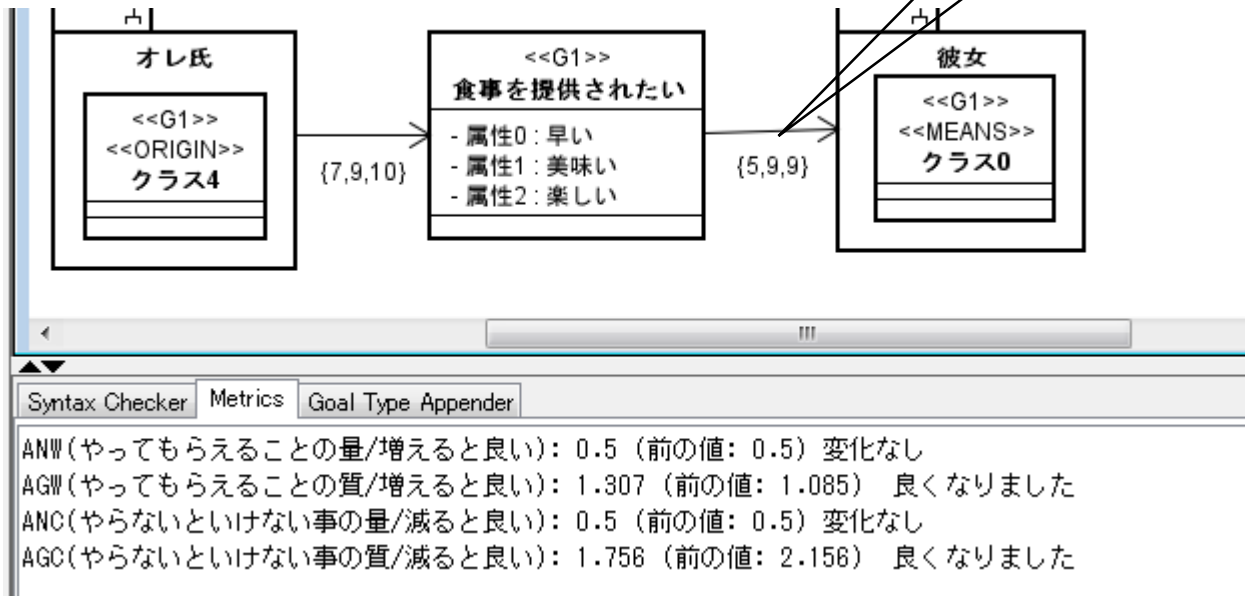
Metrics

- モデル化した活動の良さを以下の4つの観点から数値化しています.
- 数値の絶対値にはあまり意味がなく, それぞれの数値変化に着目してください.
- ANW 人がやってもらえることの平均量
- AGW 人がやってもらえることの質の平均値
 - 以上, 増えると嬉しい
- ANC 人がやらないといけないことの平均量
- AGC 人がやらないといけないことの質の平均値
 - 以上, 減ると嬉しい
- 期待や実情の数値が変化したり, やってくれるのがシステム(MACHINE)になったりすると変化します.
- 文法エラーがあると計算してくれません.

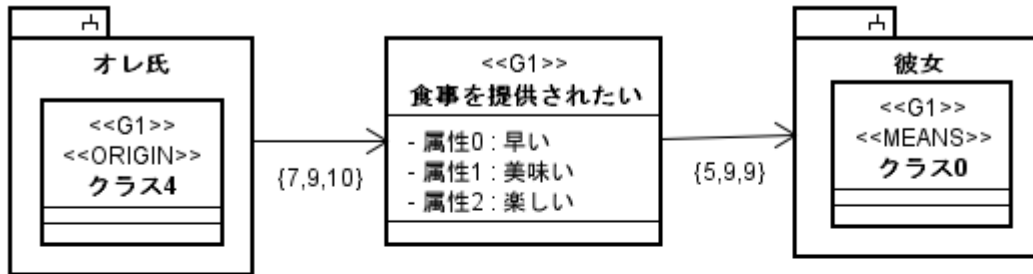
例1



彼女の料理の腕前があがって、
オレ氏の期待達成もあがり、
彼女の負担も減った。



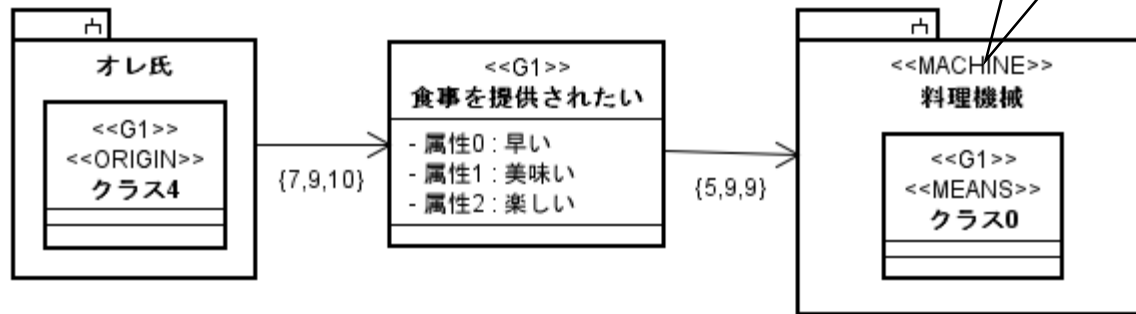
例2



G1の達成を人間である彼女ではなく、機械にやらせるようにしたので、全体によくなりました。(ただし、期待や能力は変わっていないものとしている)

Syntax Checker Metrics Goal Type Appender

ANW(やってもらえることの量/増えると良い): 0.5 (前の値: 0.5) 変化なし
 AGW(やってもらえることの質/増えると良い): 1.307 (前の値: 1.085) 良くなりました
 ANC(やらないといけない事の量/減ると良い): 0.5 (前の値: 0.5) 変化なし
 AGC(やらないといけない事の質/減ると良い): 1.756 (前の値: 2.156) 良くなりました



Syntax Checker Metrics Goal Type Appender

ANW(やってもらえることの量/増えると良い): 1.0 (前の値: 0.5) 良くなりました
 AGW(やってもらえることの質/増えると良い): 1.307 (前の値: 1.085) 良くなりました
 ANC(やらないといけない事の量/減ると良い): 0.0 (前の値: 0.5) 良くなりました
 AGC(やらないといけない事の質/減ると良い): 0.0 (前の値: 1.756) 良くなりました

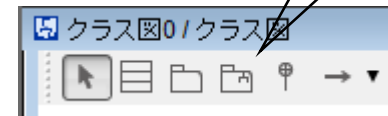
典型的なモデル化手順

1. サブシステムとしてアクターを書きます.
2. アクター間のゴールの依存関係を書きます.
3. 各ゴールで留意すべき品質項目(「早く」等)をクラスとして書きます.
 - 書いた後は画面から非表示にしてよい.
4. 各ゴールに必要な品質項目を属性として追加します.
5. 各アクター内にゴールの依存関係に応じて, ゴール階層(ゴール単体の場合もある)を書きます.
 - ORIGIN や MEANS のステレオタイプも忘れずに.
6. メトリクスを見て, 良さを確認します.
7. ファイルをコピーして, as-isモデルを修正して, to-beモデルを書きます.

TIPS

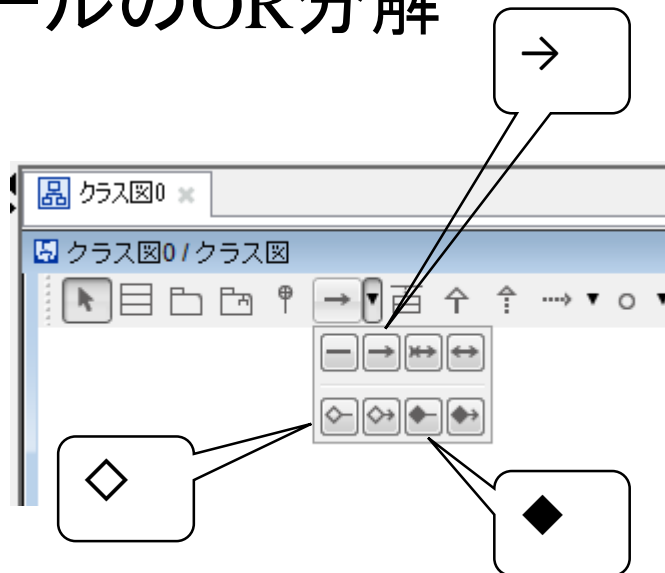
1. ステレオタイプや数値は全て半角で書く.
2. **アクター内**のクラスはデフォルトで付く「**クラス3**」等の**名前**のままにする.
 - ただし、**同じ名前**があったら、**数値の部分のみ変更**する.
 - 「クラス3」を「クラス33」にする等.
3. あるゴールに複数の品質特性があったら、矢印には、カンマで区切って、1から10の数値を特性の数分、制約として書く.
4. MEANS 等のステレオタイプも手動で付記しても無論よい.
5. サブシステムとパッケージは混同しないで！
 - パッケージだと関連が引けません.

右のを使う。



関連の種類

- ゴールを表すクラス間等の関連は、メニューから選べます。
- 基本，
- → アクターゴール間の関連
- ◆ アクター内のゴールのAND分解
- ◇ アクター内のゴールのOR分解
- しか使いません。



制約の書き方

- 関連に付記する数値は「制約」として書きます。

関係性: 開催者 (10,10) → 会議開催したい (属性0: 早く, 属性1: 正確に) → 秘書

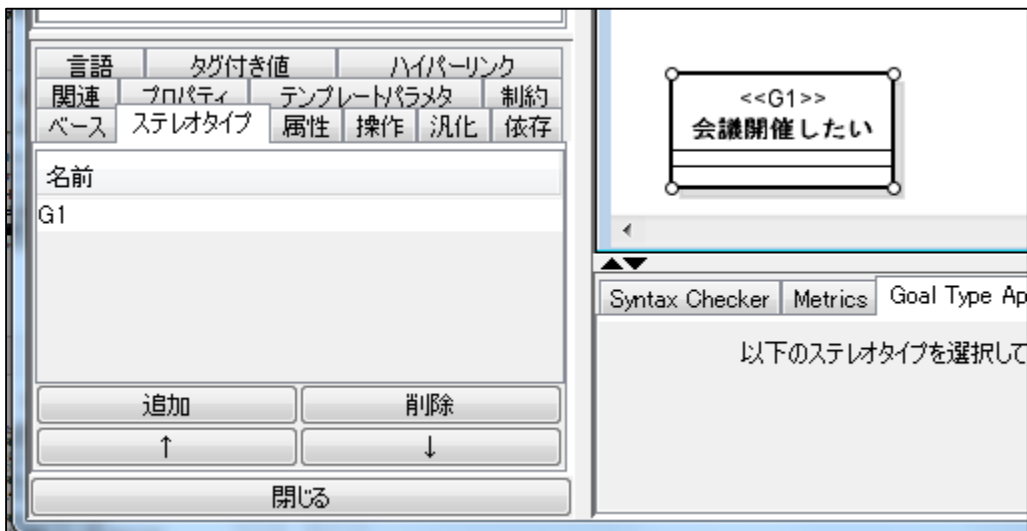
制約の編集: constraint, 3,9

関係性: 開催者 (0,10) → 会議開催したい (属性0: 早く, 属性1: 正確に) → 秘書 (3,9)

制約の編集: constraint, 3,9

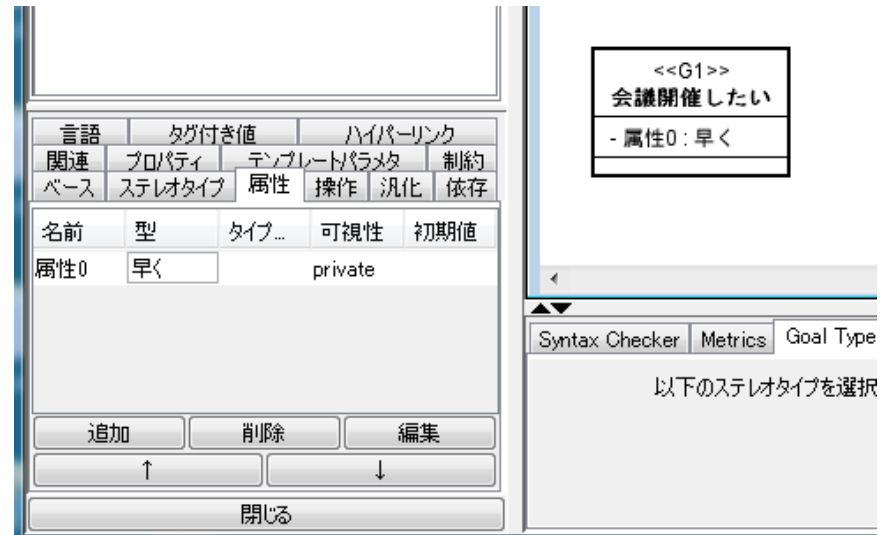
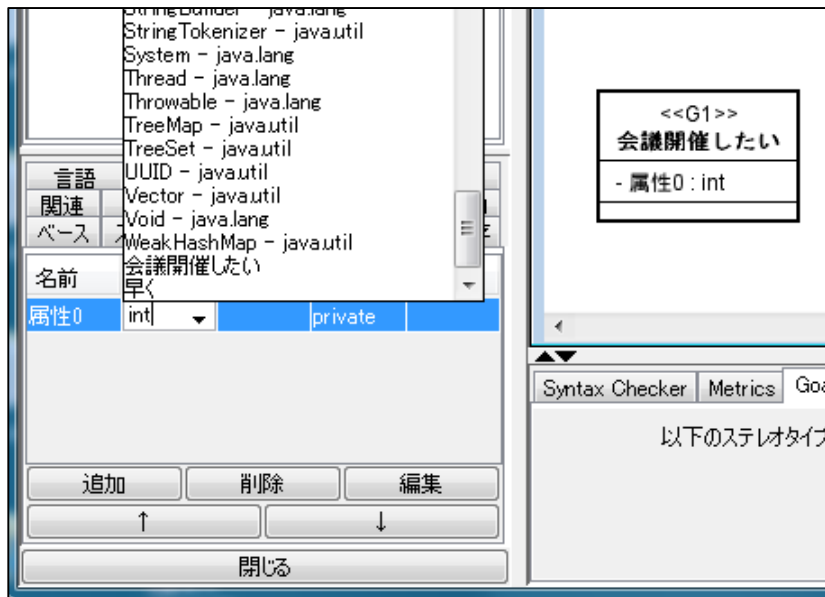
ステレオタイプの一般的な書き方

- クラス等のステレオタイプは、クラスを選択すれば、左下のステレオタイプのタブから追加、削除、修正ができます。
- これを使い、手動で、QUALITY, ORIGIN, MEANS 等をつけても、無論、OKです。



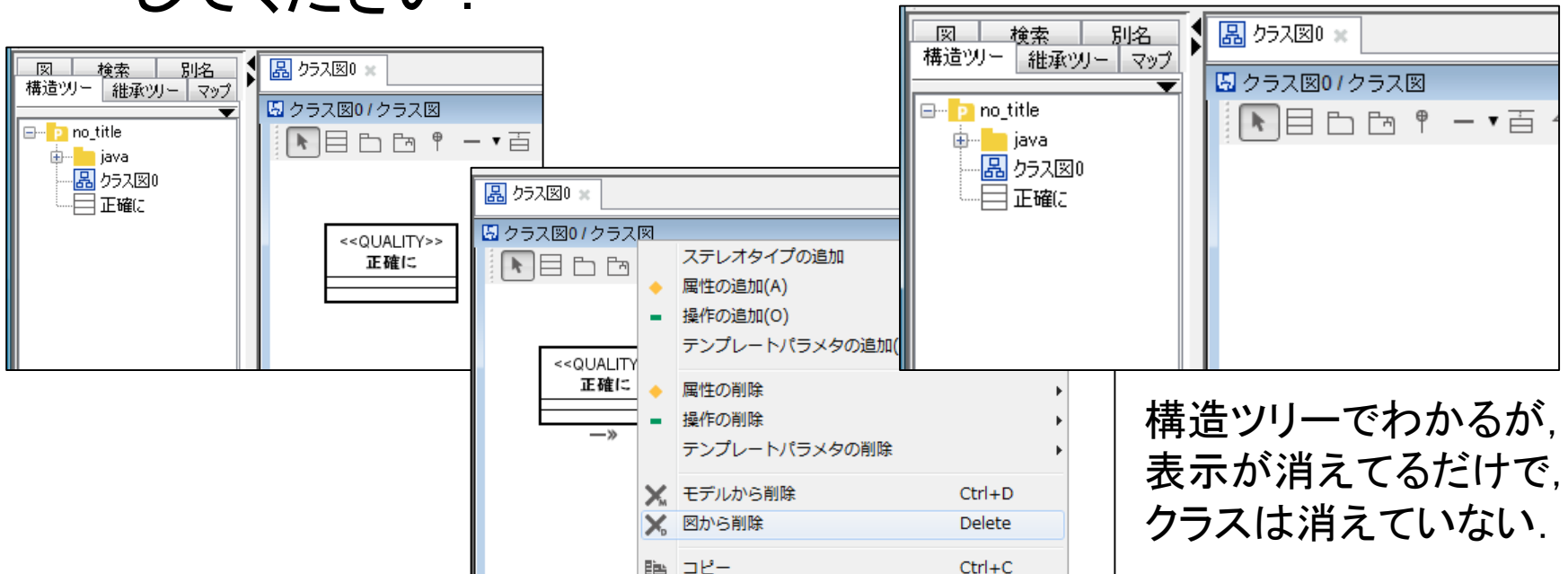
属性のつけ方

- クラスに品質特性を表す属性を付ける場合、
- クラスを選択して、
- 属性のタブを選び、
- 追加を押して、
- 型として該当する特性を選びます。
- 属性名は、変えなくてもよい。



クラスの非表示の仕方

- クラスは図上で非表示にできます。
- 削除とは意味が違います，見えてないだけです。
- QUALITYのクラス定義が邪魔な場合，非表示にしてください。



構造ツリーでわかるが，
表示が消えてるだけで，
クラスは消えていない。

選択して右クリックで，
図から削除を選択

以上