要求を整理してみよう

要求を整理してみよう

SYSML の概要を知ろう

SYSML とは

Systems Modeling Language の略でOMG（Object Management Group）によって策定されたハードウェアも含めたシステム全体を記述するためのモデリング言語です。

誕生の背景

特に組み込み分野でのハードウェアも含めたモデリングにおいて、従来のモデリング手法では問題点を抱えたままでした。特に、分析・設計において有効とされていた UML においても、仕様策定時には想定していなかった領域での問題や仕様の曖昧さ、表現力の不足といった問題が指摘されるようになりました。また、UML はあくまでソフトウェアを分析・設計する表記法で、ハードウェアなどは対象外でした。UML の配置図など一部ハードウェアを表現可能なものもありますが、ソフトウェアからの観点であり、ハードウェアそのものを表現できず、表現力の拡張が望まれていました。それを打破するべく OMG により SysML が策定され今日に至り、最新の仕様書は Version1.4 です。(2017 年 5 月 16 日現在)

SYSML の概要

SysML は、下記図のようにUML2 を拡張した仕様です。また、UML2 の全ての仕様を包含しているのではなく、一部は SysML 独自の仕様を含みます。

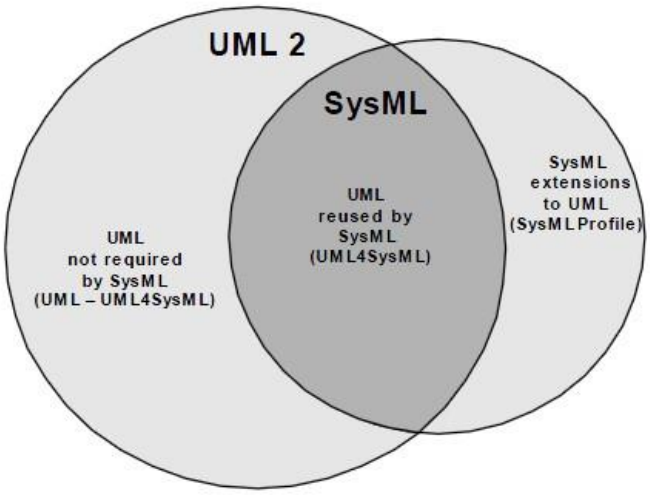


Figure 4.1 - Overview of SysML/UML interrelationship

引用：OMG SysML1.4 仕様書　P7
http://www.omg.org/spec/SysML/1.4/PDF

SYSML 図

ピンクの点線で囲まれた要求図(Requirement Diagram),パラメトリック図(Parametric Diagram)が新規図で、太枠のアクティビティ図が UML2 の拡張、緑文字のブロック定義図(Block Definition Diagram)は、UML2 クラス図の拡張の新規図、内部ブロック図(Internal Block Diagram)は UML2 の合成構造図の拡張の新規図です。

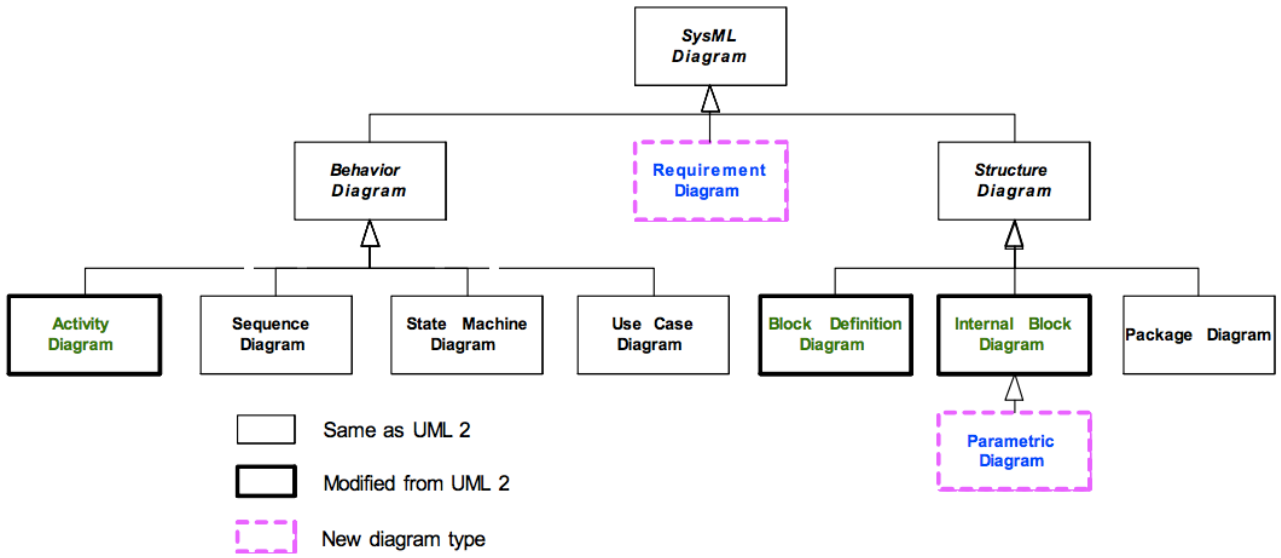


Figure A.1 - SysML Diagram Taxonomy

astah*が持つ要求周りの機能

astah* professional、SysML では、SysML の要求に関連した部分を、ソフトウェア開発の現場で汎用的に使用できるよう実現しました。要求およびテストケースの階層・派生関係の設定、要求テーブルを用いた管理を容易にし、工程間のギャップを減らします。

要求

システムの要求を記述するモデルです。名前、ID、テキスト等を設定でき、階層をツリーで表現します。また、導出、コピー、満足、検証、洗練、トレースといった派生関係を設定できます。その他の機能として、要求からユースケース、マインドマップトピック等への変換もサポートしています。

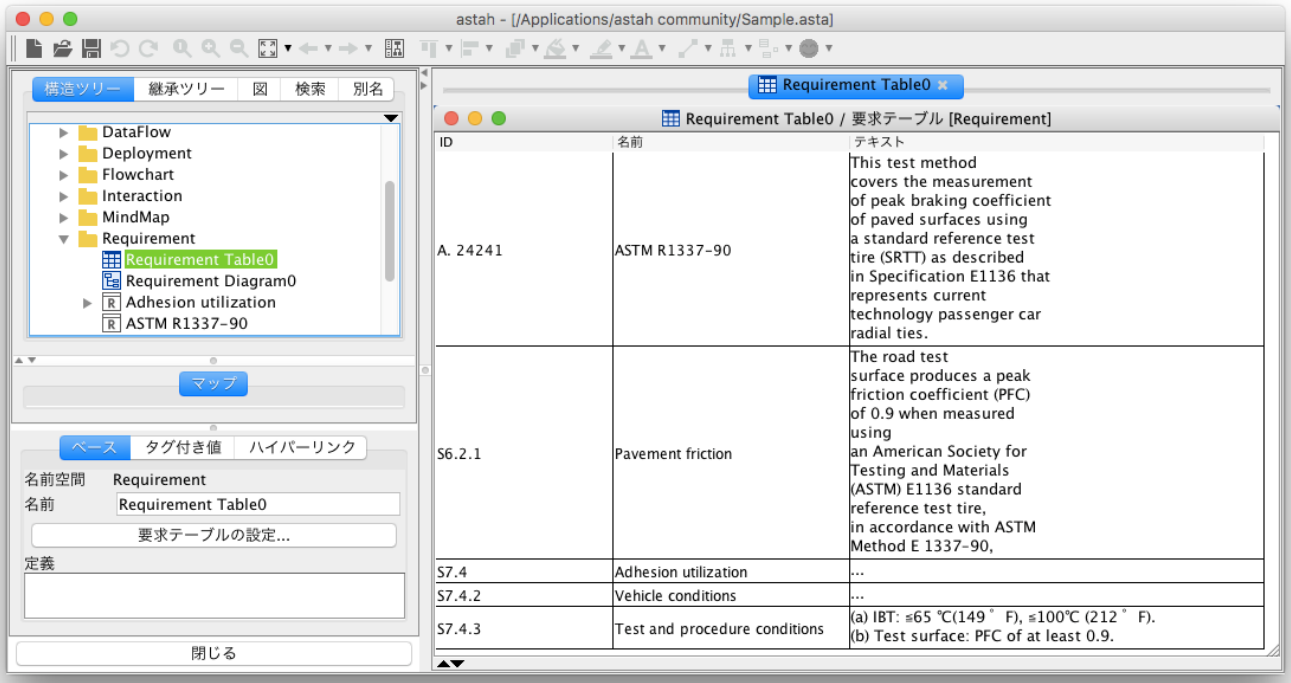
テストケース

システムのテストケースを記述するモデルです。名前、ID、定義等を設定でき、階層をツリーで表現します。また、満足、検証、洗練といった、テストケースからの派生関係も設定可能です。テストケースからユースケース、マインドマップのトピックへの変換もサポートしています。

要求テーブル

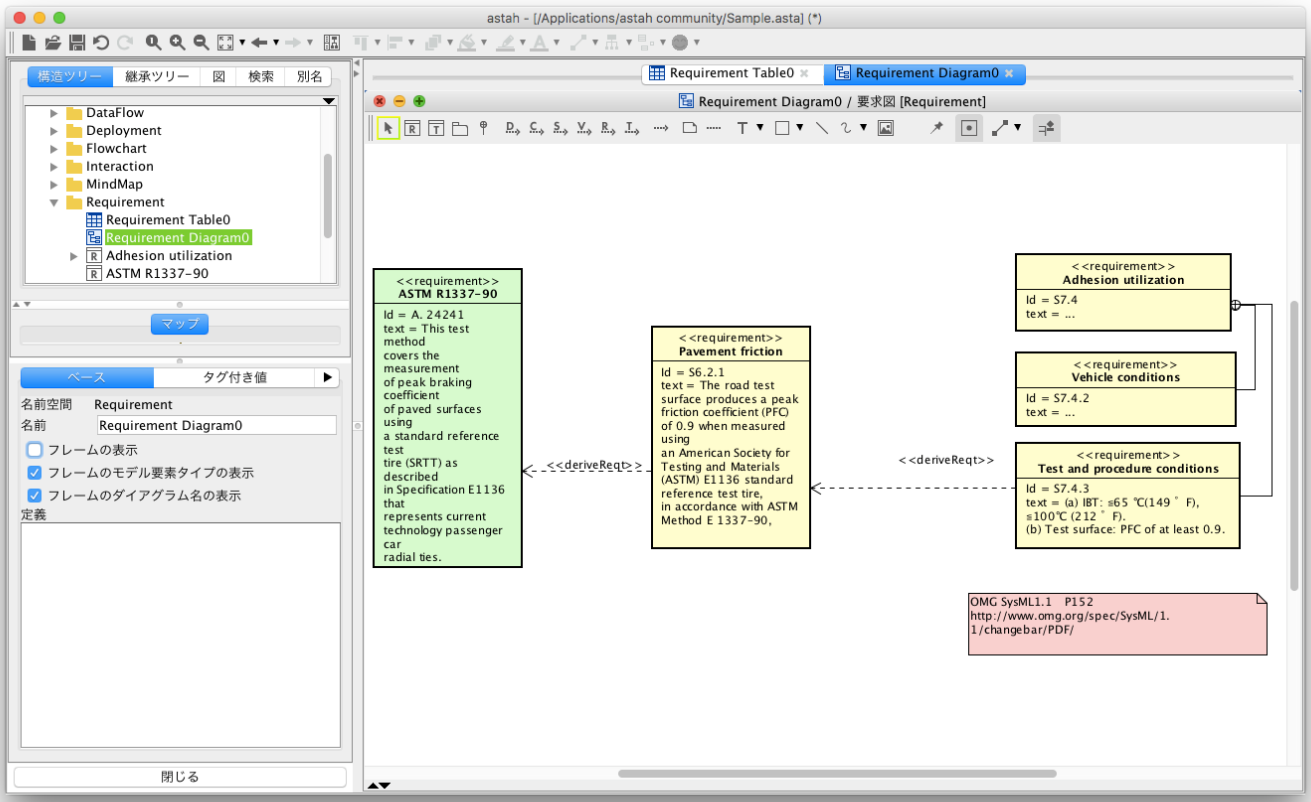
要求を表形式で編集したり、階層を指定して要求のID、名前、テキストを表示できます。Excel の入出力も可能です。

デモ動画：https://www.youtube.com/watch?v=Dyj8SVSPqgE&feature=youtu.be



要求図

どんな要求が存在するか、また要求間の関係を定義する図です。
デモ動画：https://www.youtube.com/watch?v=Dyj8SVSPqgE&feature=youtu.be



導出、コピー、満足、検証、洗練、トレース

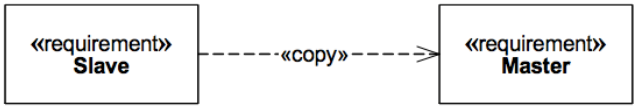
導出 <<DERIVEREQT>>

Derive Dependency		**SysML::Requirements:: DeriveReqt**

導出は、クライアント要求がサプライヤ要求に導き出される二つの要求間の依存関係です。例えば、システム要求はビジネス要求に由来するかもしれません。あるいは、より低レベルの要求は、システム要求に由来するかもしれません。矢印方向は、クライアント要求から、それが導き出されるサプライヤ要求までを指します。

astah*では、要求から要求に引くことができます。

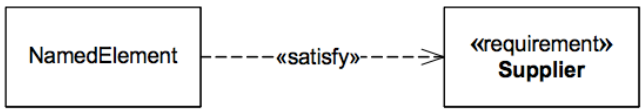
コピー <<COPY>>

Copy Dependency		**SysML::Requirements:: Copy**

コピーは、クライアント要求とサプライヤ要求間の依存関係で、クライアント要求のテキストがサプライヤ要求の読み取りコピーであることを示します。異なる状況で再利用する目的で、クライアント/サプライヤ関係を維持します。クライアント要求は、コピーの矢印の先のサプライヤ要求の読み取り専用コピーです。

astah*では、要求から要求に引くことができます。

満足 <<SATISFY>>

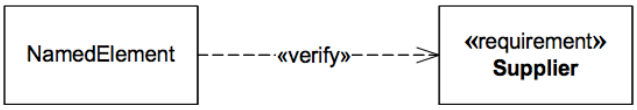
Satisfy Dependency		**SysML::Requirements:: Satisfy**

満足は、要求とその要求を満足させるモデル間の依存関係です。他の依存関係と同様に、矢印方向はクライアントモデルからそれを満たすサプライヤ要求まで指します。

astah * では、満足を、下記モデルから要求に引くことができます。

パッケージ 、モデル 、サブシステム 、クラス 、関連クラス、インターフェース、エンティティ、コントロール 、バウンダリ 、アクター、 ユースケース、コンポーネント 、成果物 、ノード 、要求 、テストケース

検証 <<VERIFY>>

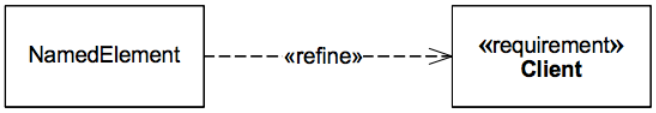
Verify Dependency		**SysML::Requirements:: Verify**

検証は、要求とシステムが要求を満たすかどうかを検証するテストケースの依存関係です 他の依存関係と同様に、矢印

方向は、クライアントテストケースからそれが導き出されるサプライヤ要求まで指します。

astah*では、テストケースから要求に引くことができます。

洗練 <<REFINE>>

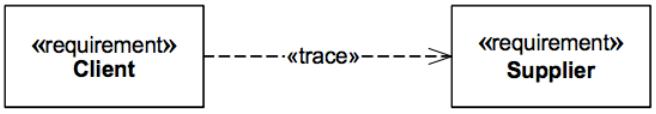
Refine Dependency		**UML4SysML::Refine**

洗練とは、モデルから要求に詳細化する依存関係です。他の依存関係と同様に、矢方向は、クライアントモデルか らサプライヤ要求の方向を示します。

astah＊では、満足を、下記モデルから要求に引くことができます。

パッケージ 、モデル 、サブシステム 、クラス 、関連クラス、インターフェース、エンティティ、コントロール 、バウンダリ アクター、ユースケース、コンポーネント 、成果物 、ノード 、要求 、テストケース

トレース <<TRACE>>

Trace Dependency		**UML4SysML::Trace**

トレースとは、要求間の抽象的なつながりを表す依存関係です。

astah*では、要求から要求に引くことができます。

要求図と要求テーブルを使用して要求をモデリングしよう

以下は、OMG から提供されているSysML1.1 仕様書の要求図のサンプルで、
[組み込み系サンプル]で自動車の舗道摩擦まわりの要求を表したものです。

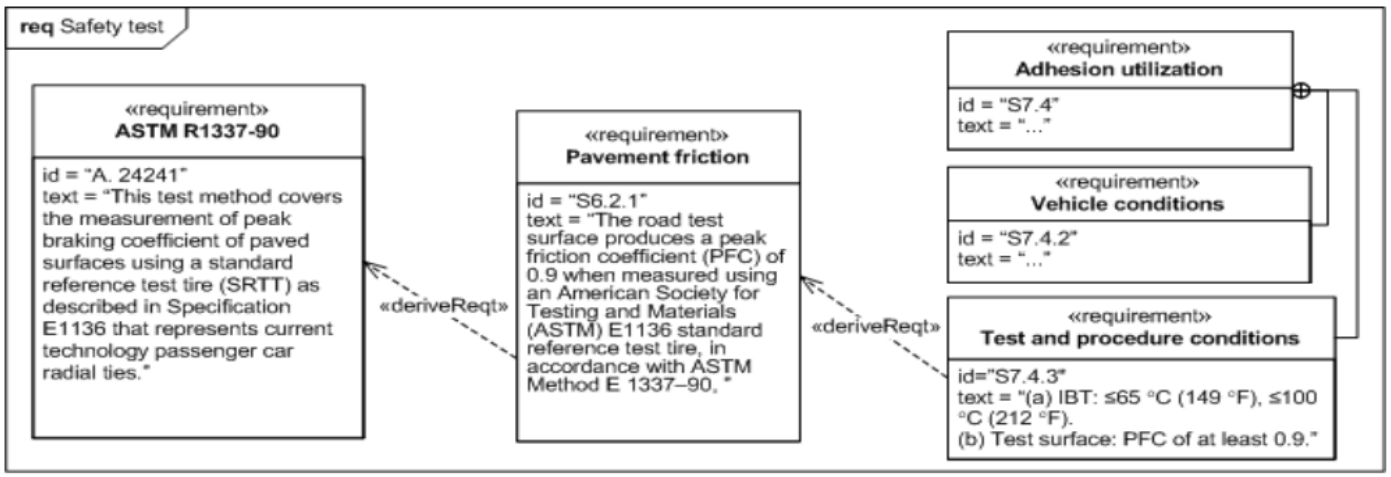
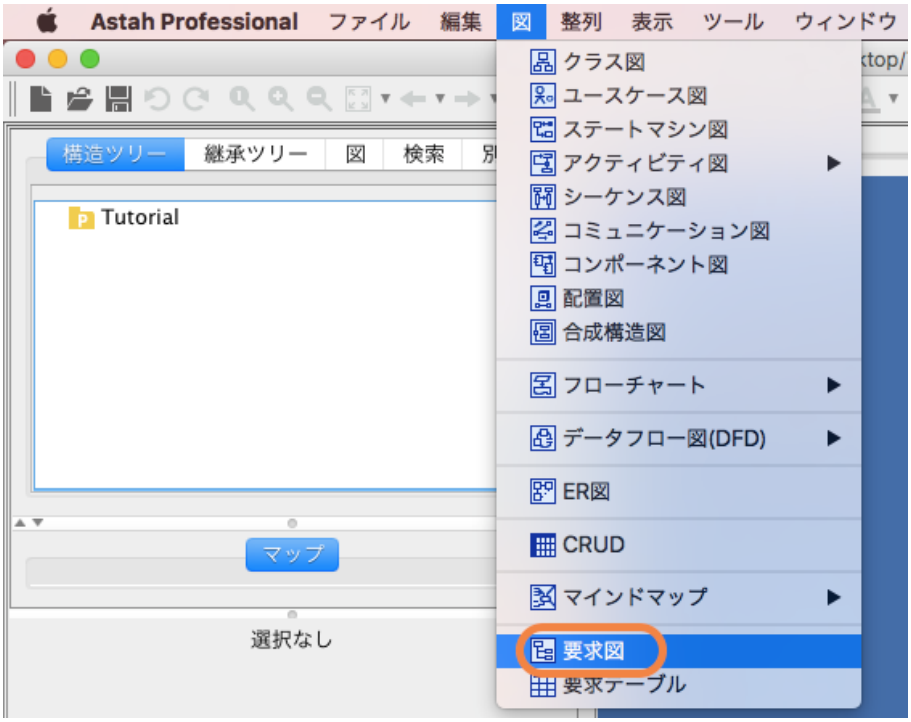


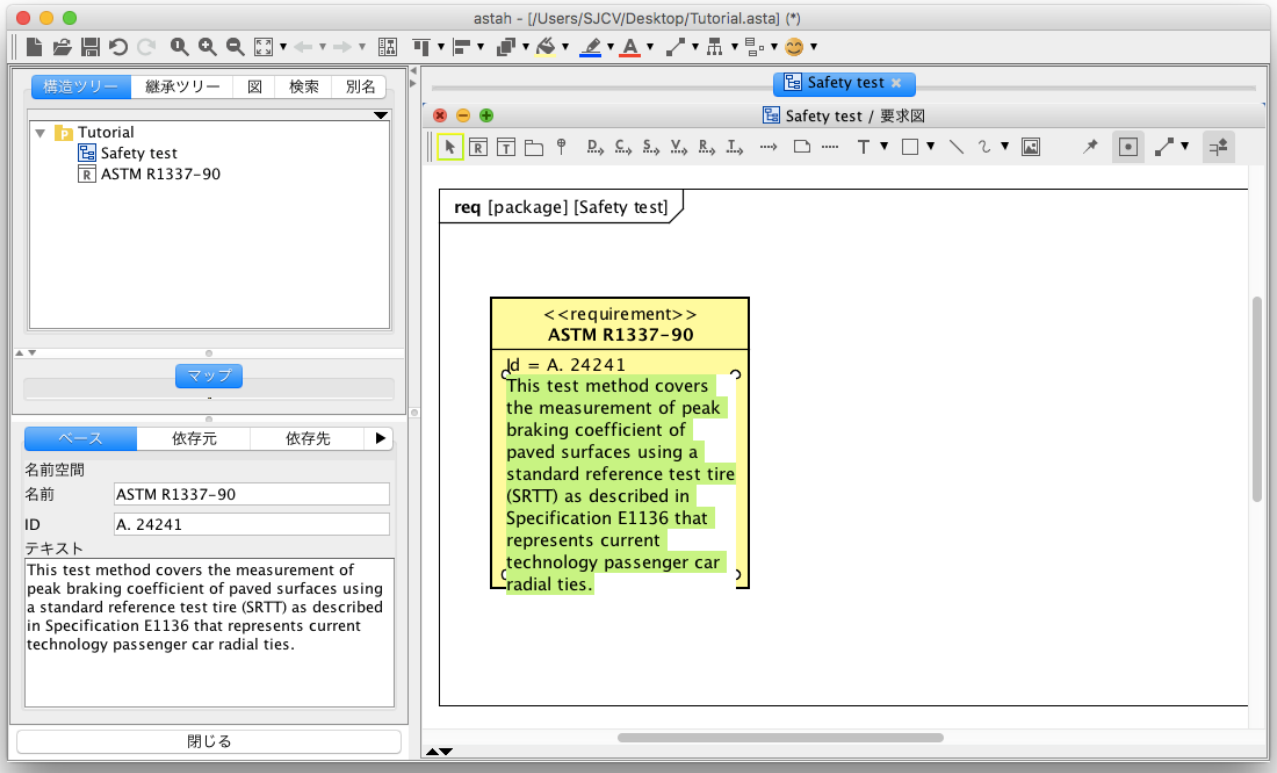
Figure 16.2 - Requirements Derivation

これと同等のモデルを astah*で描いていきましょう。

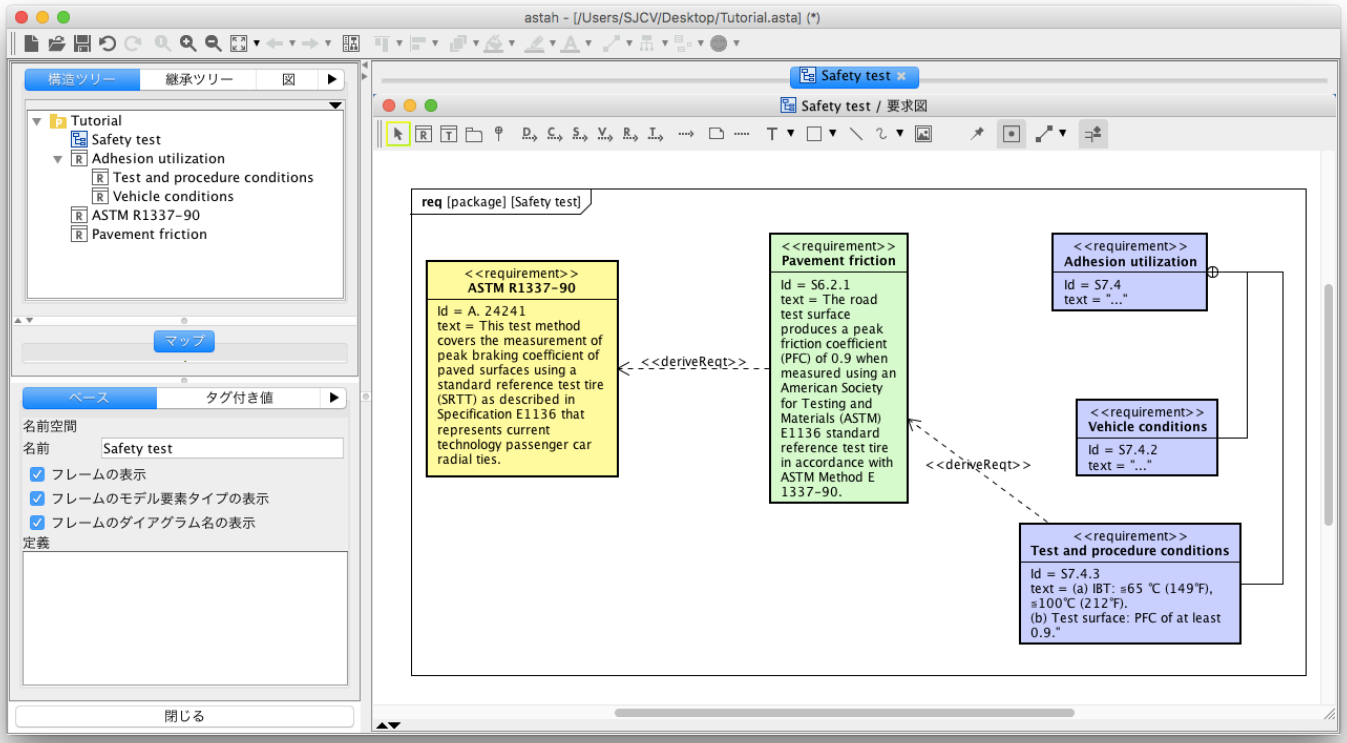
まず、図メニューから“要求図”を選択し、要求図を作成します。



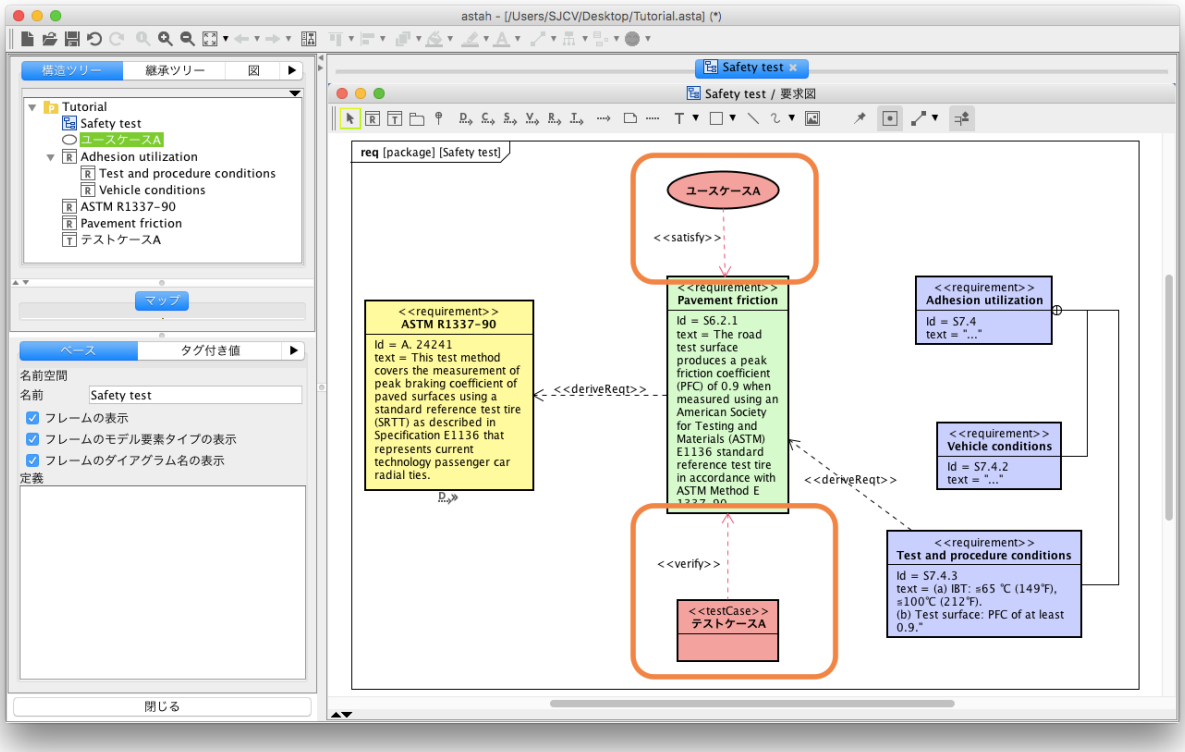
図上をダブルクリックして要求を作成します。要求の名前、ID を入力します。Text の項目には、その要求の説明を入力します。



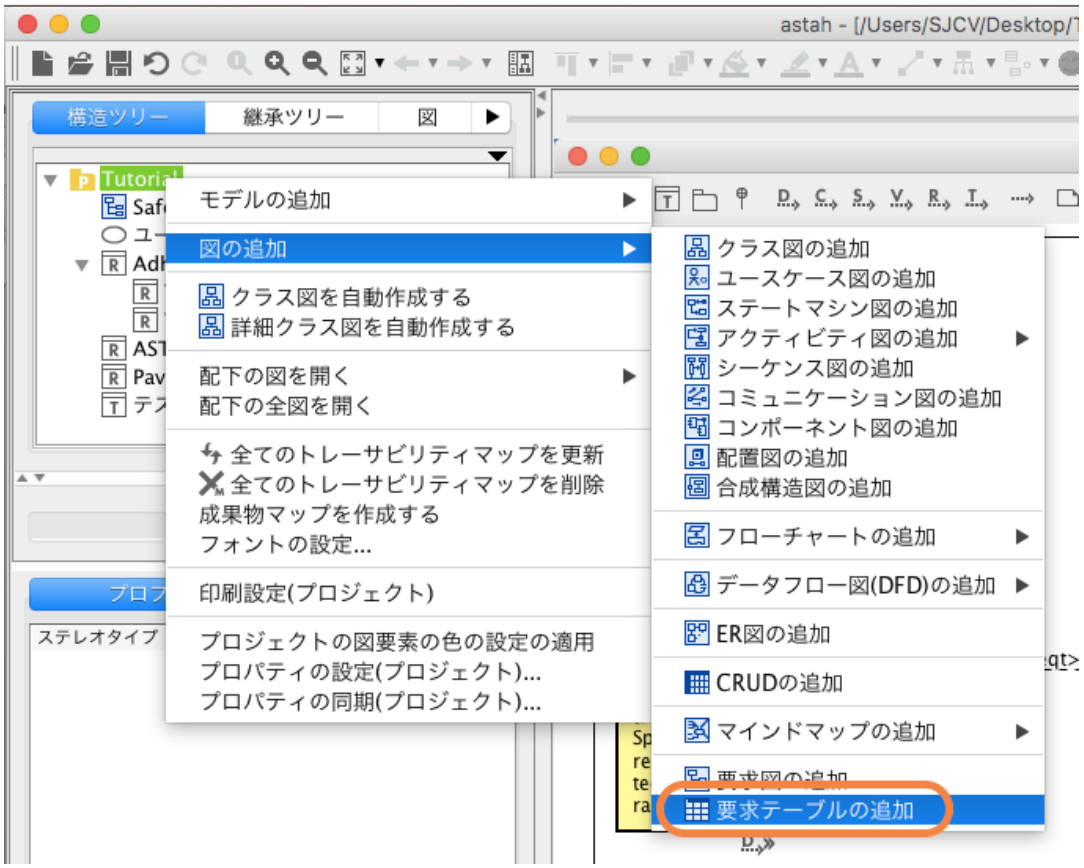
同様に他の要求についても作成します。次に、要求間の関係を作成してみましょう。要求間の導出関係とネスト関係をそれぞれ、ツールバーのボタンを押して、作成します。ネスト関係については、作成時に、親子関係が変わることを、構造ツリーで確認できます。ここまでで、例の要求図が完成しました。図だと要求間の関係を一目で把握できますね。



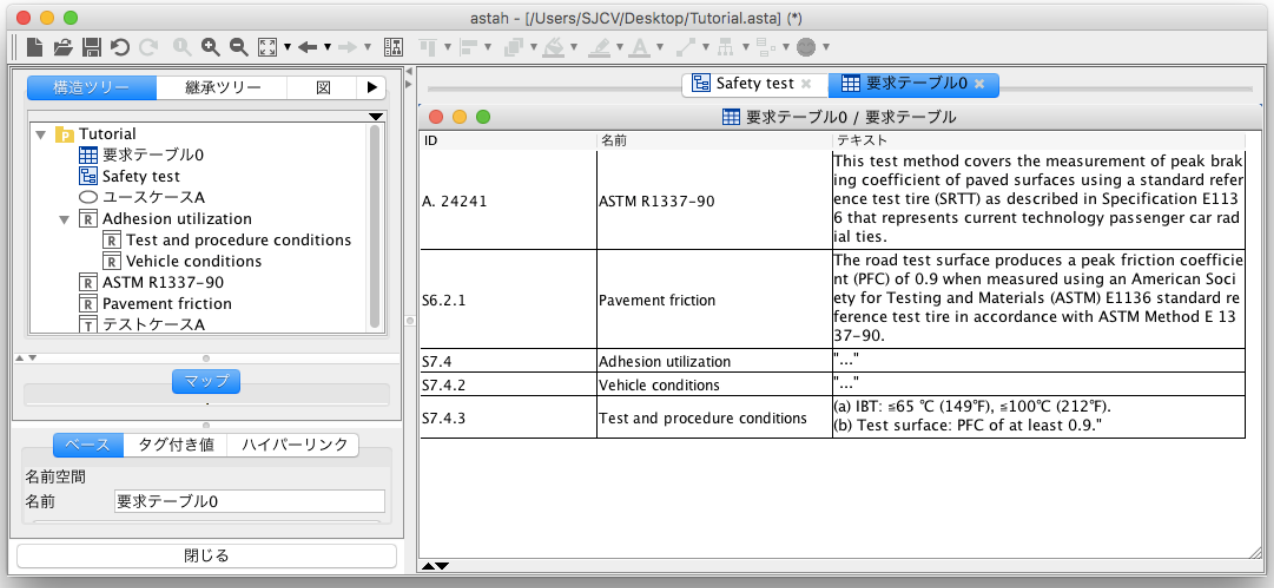
これらの要求と、分析・設計モデルとの関係も定義してみましょう。具体的な例ではないですが、もし[ユースケース A]が、[Pavement friction]を満足する関係があり、[テストケース A]が[Pavement friction]を検証する関係である場合は、次のように図上で表現することができます。ユースケースは、構造ツリーから要求図にドラッグ＆ドロップして配置することができます。



また、下のようにパッケージを指定して、要求テーブルを作成することもできます。

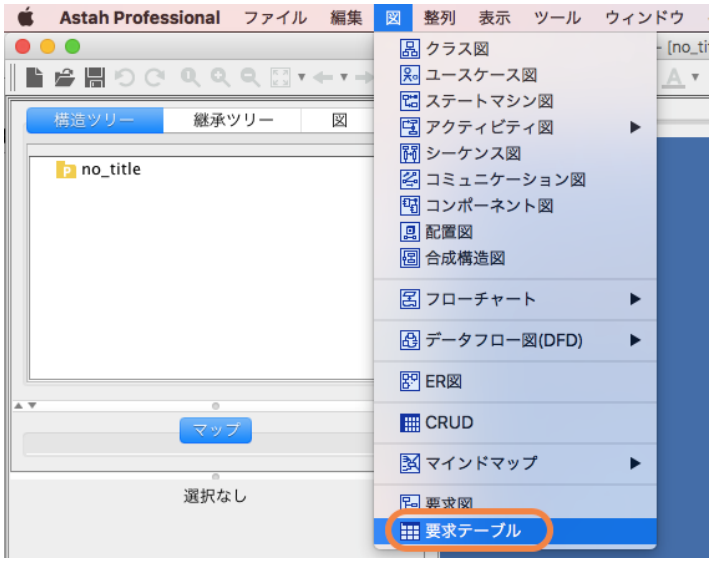


要求テーブルは、ネームスペース毎に 1 図作成できます。要求テーブルには、そのネームスペース配下に存在する要求が自動的にリストに表示されます。

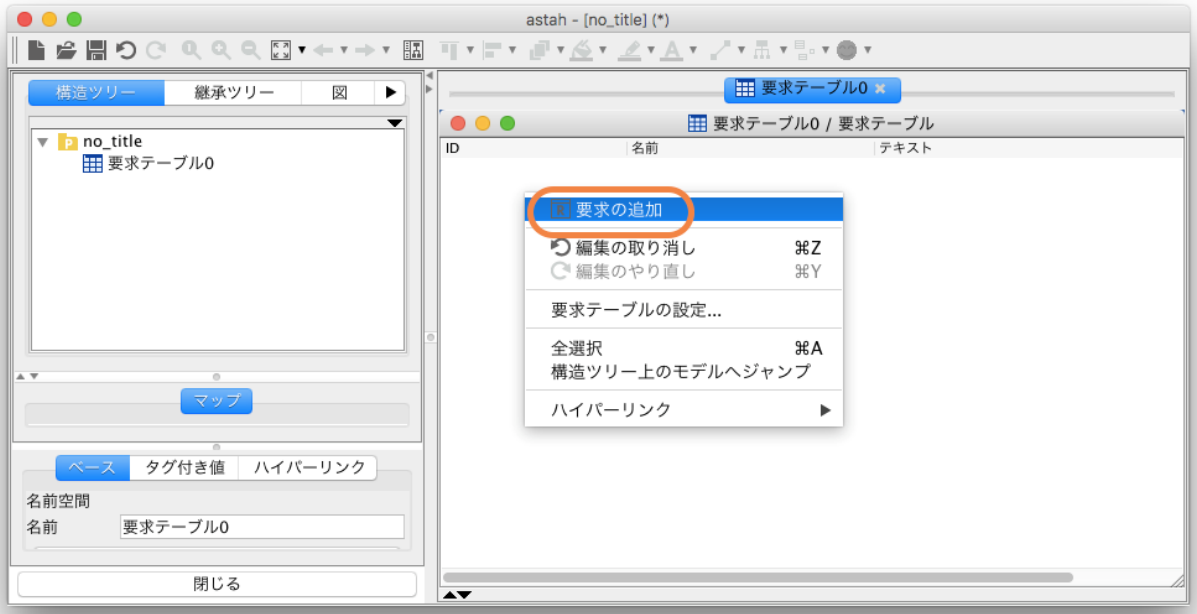


要求図と要求テーブルでは同一のモデルが利用されますので、一方の変更は、もう一方に反映されます。
また、要求テーブル上の要求のポップアップメニューからは、要求図上の対象要求へジャンプできるメニューがあります。

さて、次は、ここまで作成した要求を、要求図からではなく、要求テーブルから作成する方法を見てみましょう。まず、新規に要求テーブルを作成します。図メニューから[要求テーブル]を選択し、要求テーブルを新規作成します。



要求テーブル上を右クリックしてポップアップメニューを開き、[要求の追加]を選択すると、要求を作成できます。



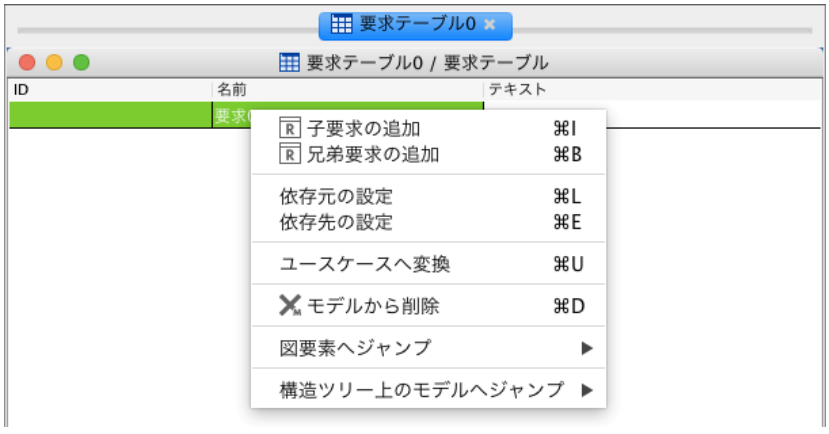
要求テーブルからは要求のID、名前、テキスト編集のほか、以下の操作も可能です。

・子要求の追加

・兄弟要求の追加

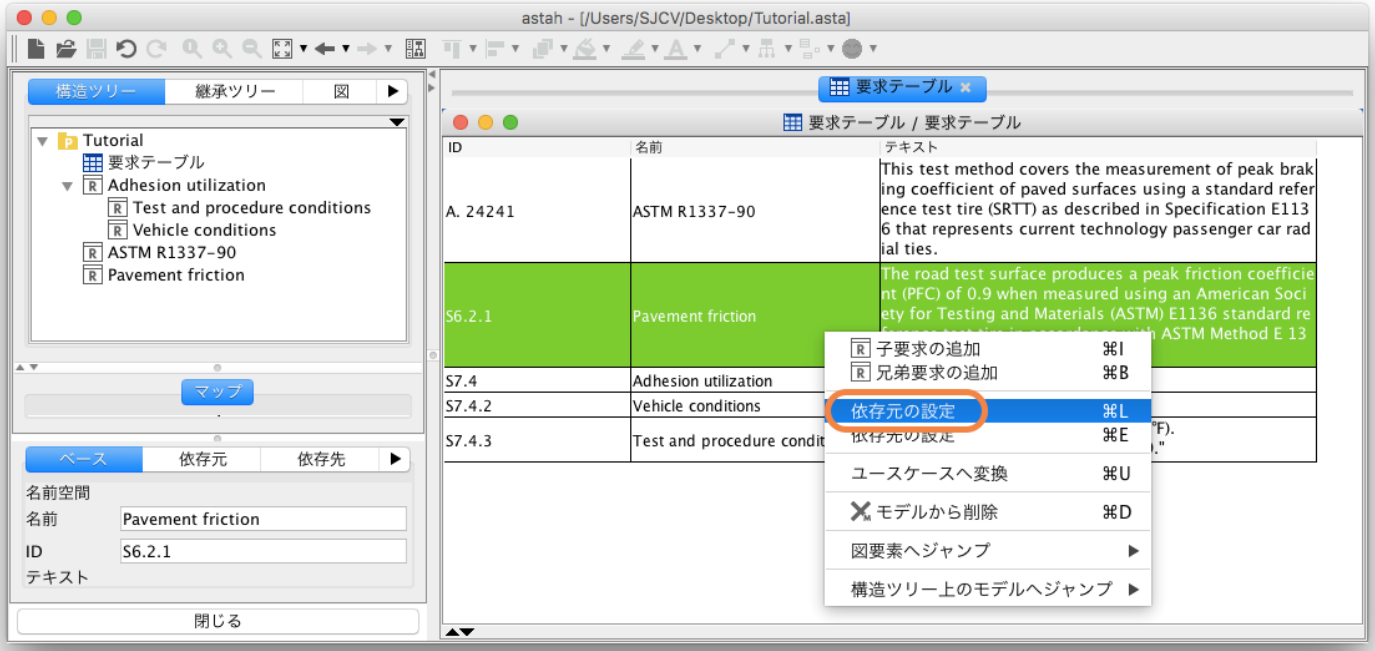
・依存元の設定

・ユースケースへの変換

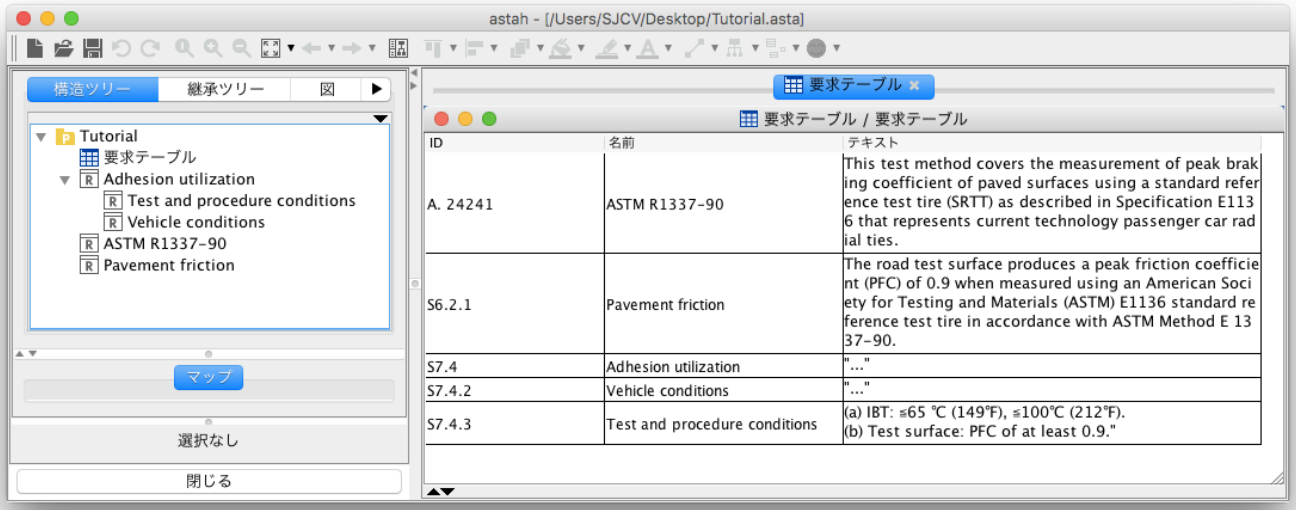


Vehicle conditions は、Adhesion utilization とネスト関係がありますので、Adhesion utilization のポップアップメニューを開き[子要求の追加]から Vehicle conditions を作成します。Test and procedure conditions も同様です。

また、Test and procedure conditions からPavement friction、Pavement friction からASTM R1337-90 は、**導出<<deriveRept>>**でつながっていますので、各要求のポップアップメニュー[依存先の設定]から設定します。([依存元の設定]でも可能です) ちなみに、導出<<deriveRept>>とは、要件から別の要件が導き出される関係のことです。



以下が作成したモデルです。



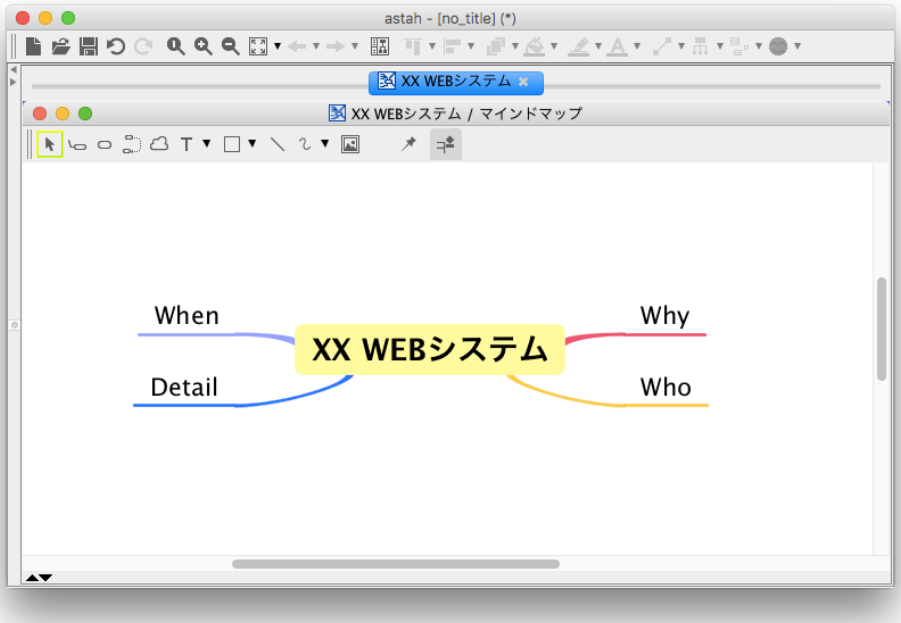
要求テーブルと要求図の両方を使って、それぞれの利点を活かすといいですね。

要求図と要求テーブルを使用して要求モデリングしよう

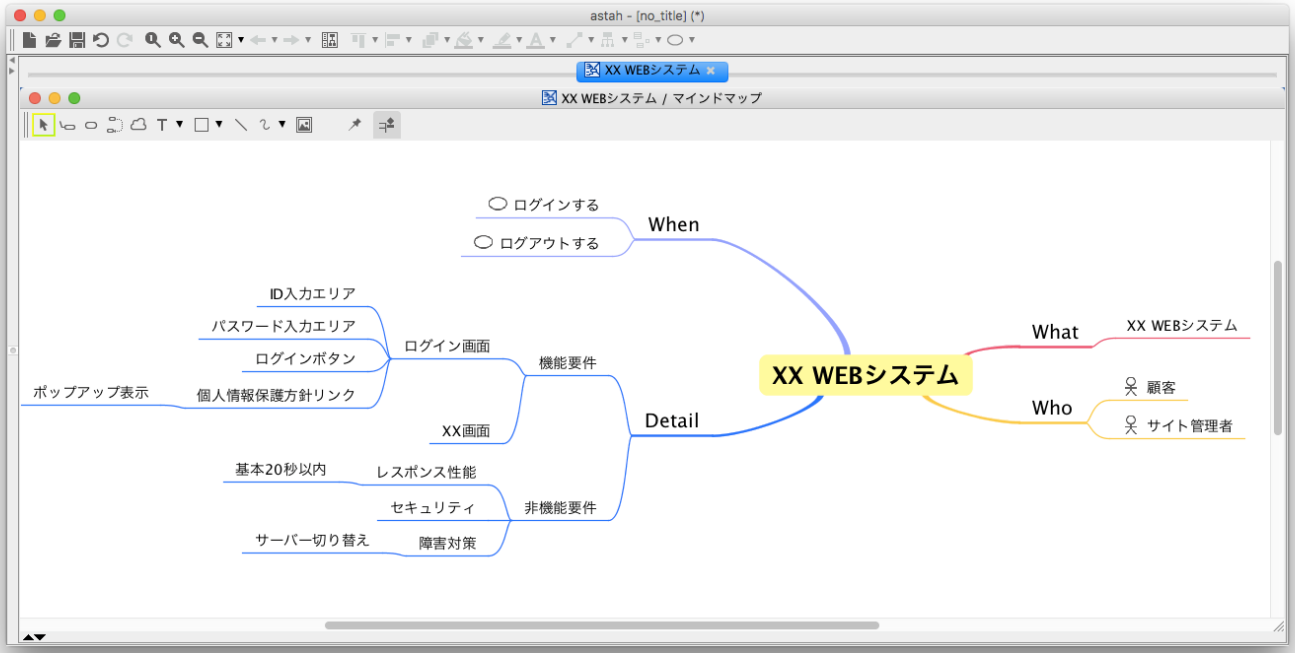
ここでは、ストーリー仕立てで要求モデリングしていきたいと思います。

登場人物	
プロジェクトマネージャー	A さん
営業	X さん
開発リーダー	B さん
要求開発リーダー	R さん

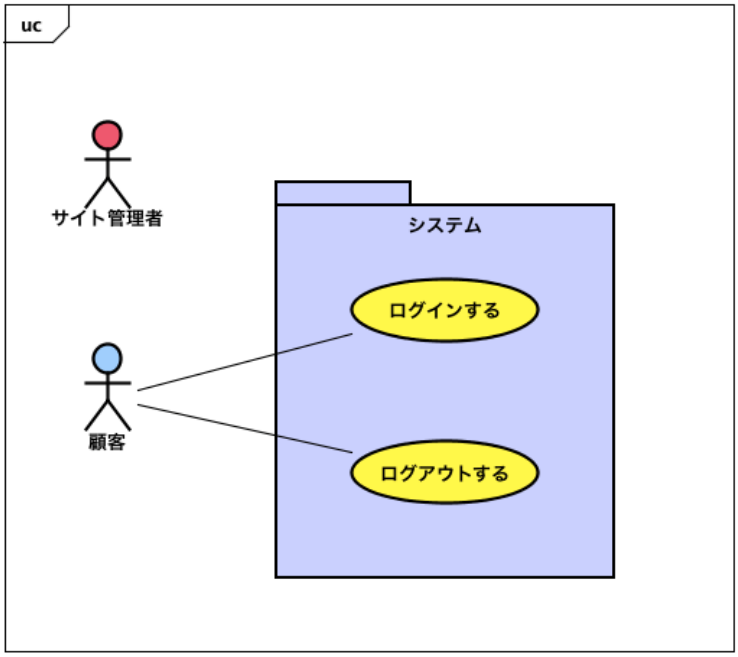
プロジェクトマネージャーであるA さんは、営業X さんと取引先に新案件の打ち合わせに行きました。
A さんは、いつものようにPC を起動し、下記テンプレートのマインドマップを使ってヒアリングを開始しました。



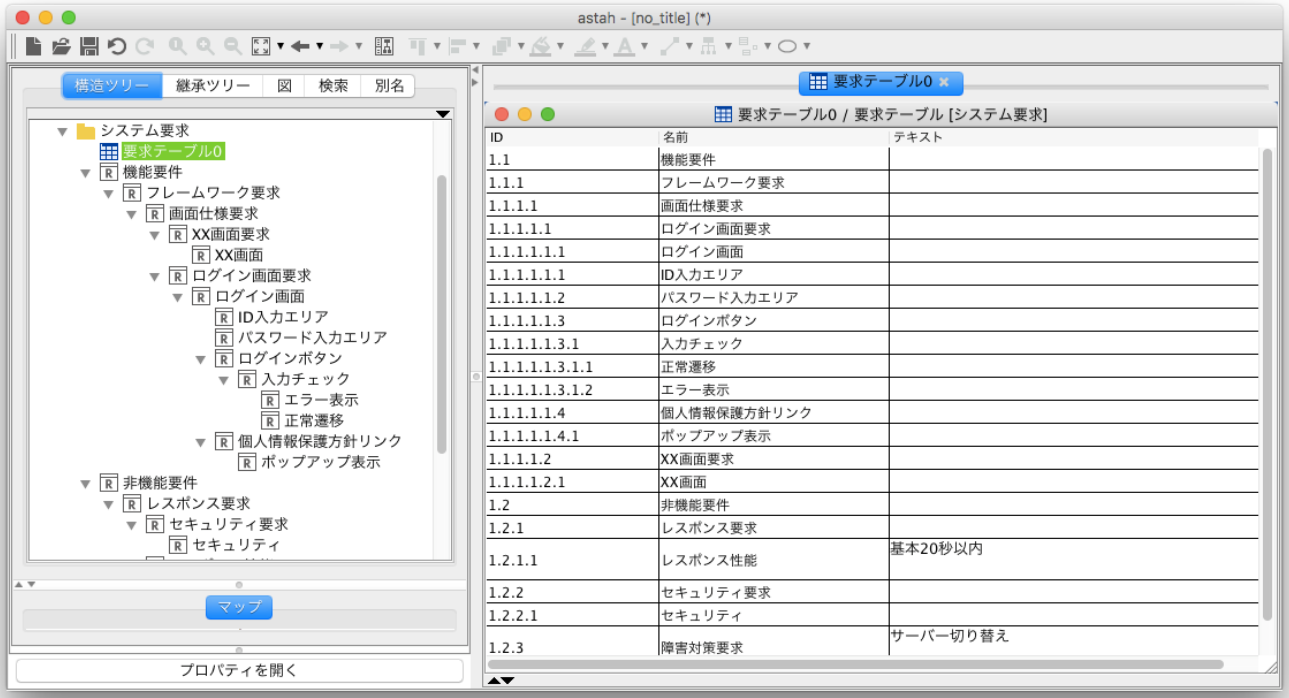
顧客の依頼は、ある WEB アプリケーションの構築です。いつもおざなりになりがちな非機能要件周り、レスポンスやセキュリティ、障害対策について注意してほしいという事を強く伝えられました。機能要件については、ログイン程度の簡単なものにとどまり、打ち合わせも無事終了しました。作成したマインドマップは以下です。



プロジェクトマネージャーA さんは、会社に戻り、さっそく以下なユースケースを作成しました。



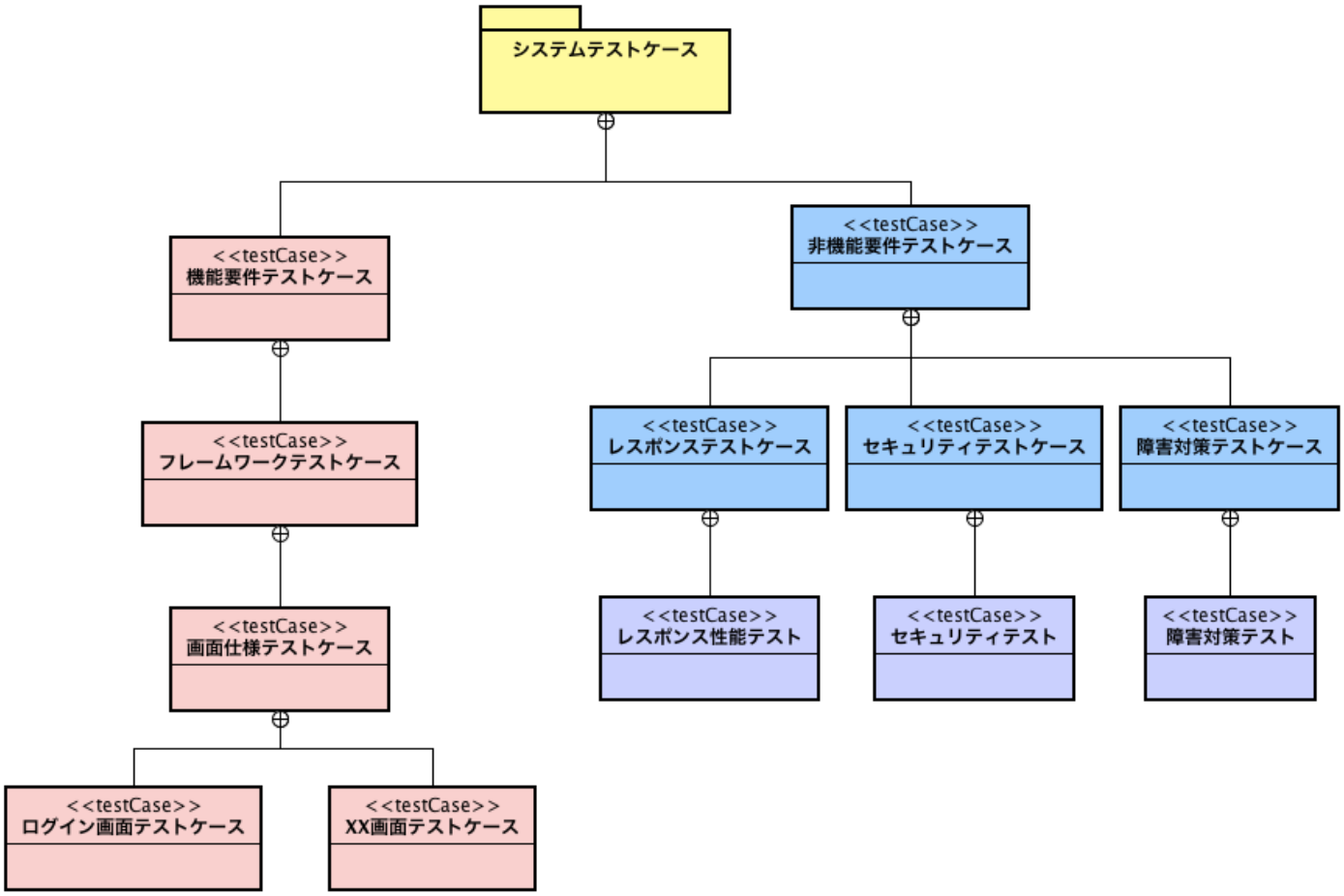
その後、A さんは、ヒアリングした要求を、機能要件、非機能要件も含めてまとめてみることにしました。頭のなかに浮かんだのは以下のモデルです。A さんは、要求テーブルからネスト関係を考慮しながら、要求を入力していきました。



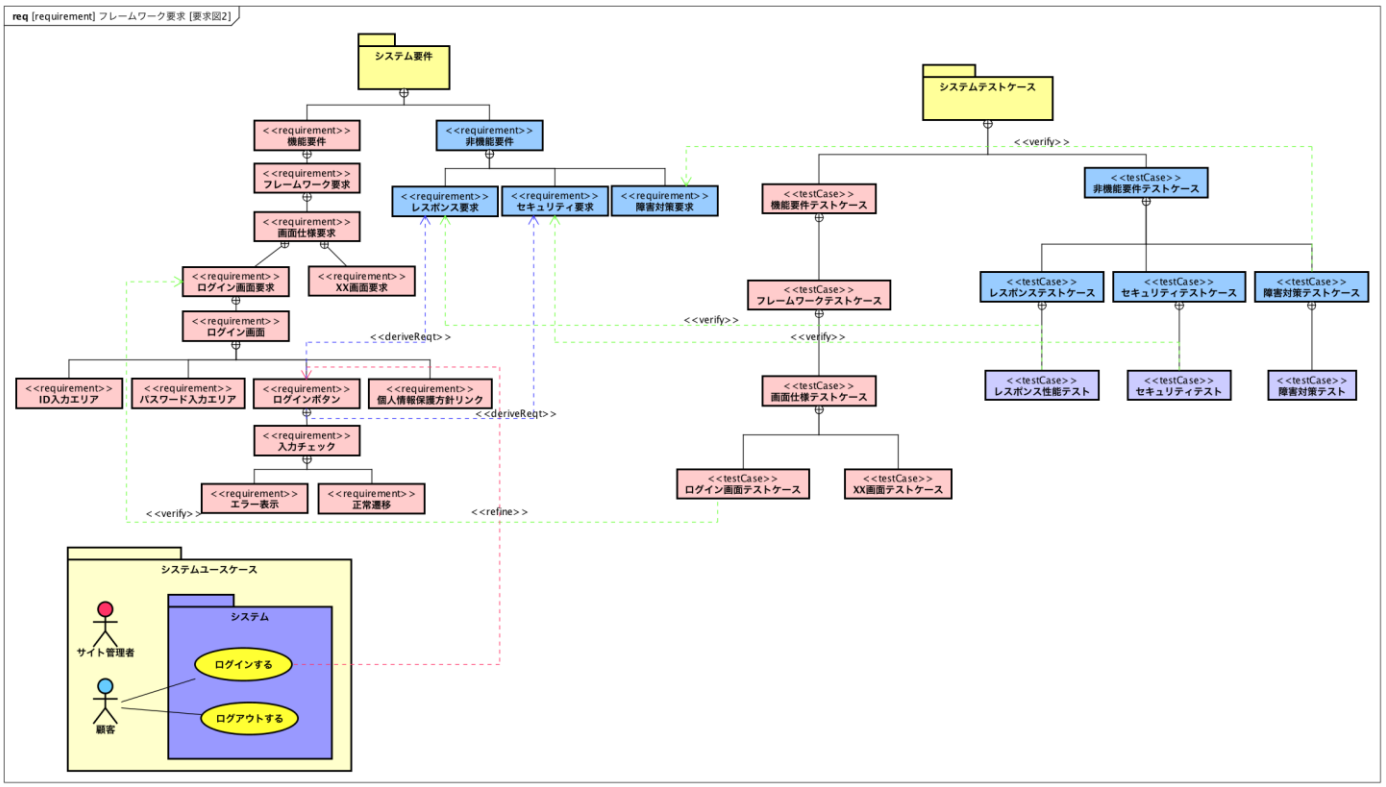
次に、要求間の依存関係を設定していきます。先ほど軽くユースケース分析をしました。ユースケースの[ログインする]の詳細は、要求の[ログインボタン]にあたりますので、機能要件のユースケースから、機能要件の要求に対して詳細化するという意味合いの**洗練<<refine>>**を引きましょう。要求は、このように既存の UML 図と関連し合っているため、要求と機能を可視化することは、仕様の共有化に一役買うことにもなるでしょう。
洗練<<refine>>は、要求テーブル上で要求を選択し、ポップアップメニュー[依存先の設定]から設定します。

A さんは、顧客からログイン時のレスポンスやセキュリティの要求に注意するよう言われていた事を思い出しました。非機能要件を満たせなければ、顧客からの信頼を失いかねません。機能要件の要求[ログインボタン]は、非機能要件の要求[レスポンス要求]、[セキュリティ要求]を導き出しています。機能要件の要求[正常遷移]から[XX 画面要求]も同様です。**導出<<deriveRept>>**は、要件から別の要件が導き出される関係のことを表すので、この関係も[依存先の設定]から設定します。

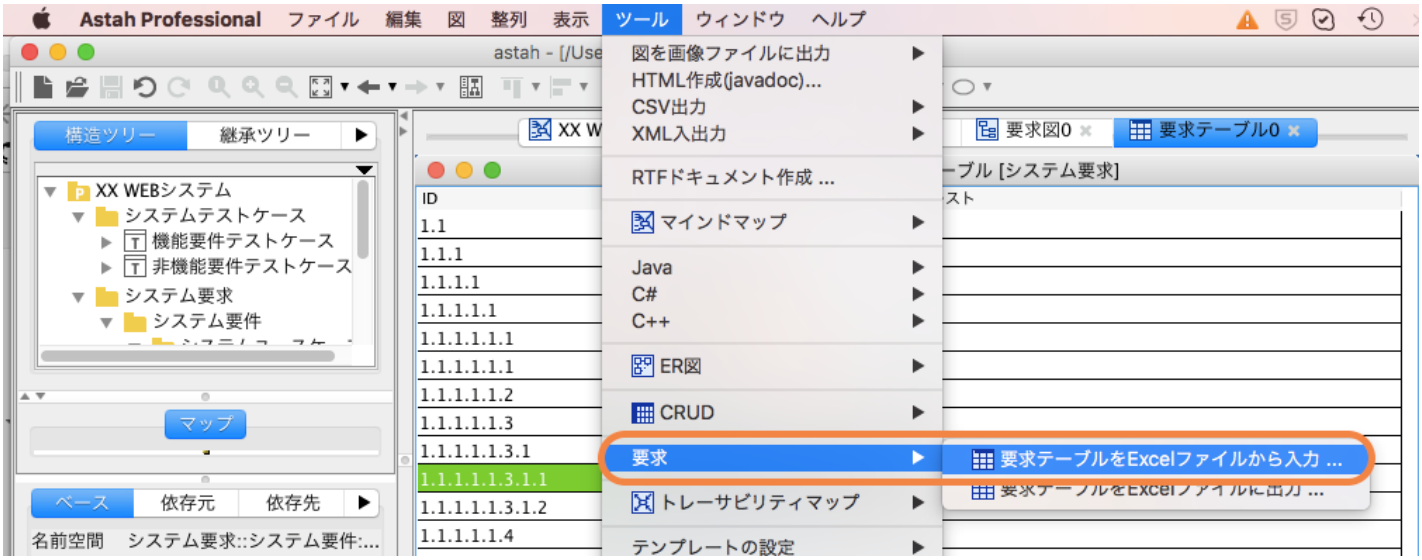
次に、テストケースについて考えます。テストケースは、astah*内でもモデルとして生成できます。
実際のテストケースの内容については、自動テストにするか、EXCEL でテスト仕様書を書くのかが決まっていません。次回のミーティングで、顧客や開発チームと話し合うため、成果物を何にするか、とういことは、懸案事項として置いておくことにします。ただ、自動テストであれば、ソースや結果のドキュメント。テスト仕様書なら、ドキュメント。シナリオならフローチャートやアクティビティ図へのハイパーリンクを張る、またはテストケースの定義に書くなど、事前にチーム内で話し合い、よりよい合意をとったほうがよさそうです。
A さんは、この点をメモしておき、開発リーダーB さんに、テストケースの設計を指示しました。B さんは、ネスト関係に注意しながら、テストケースの設計を始めました。



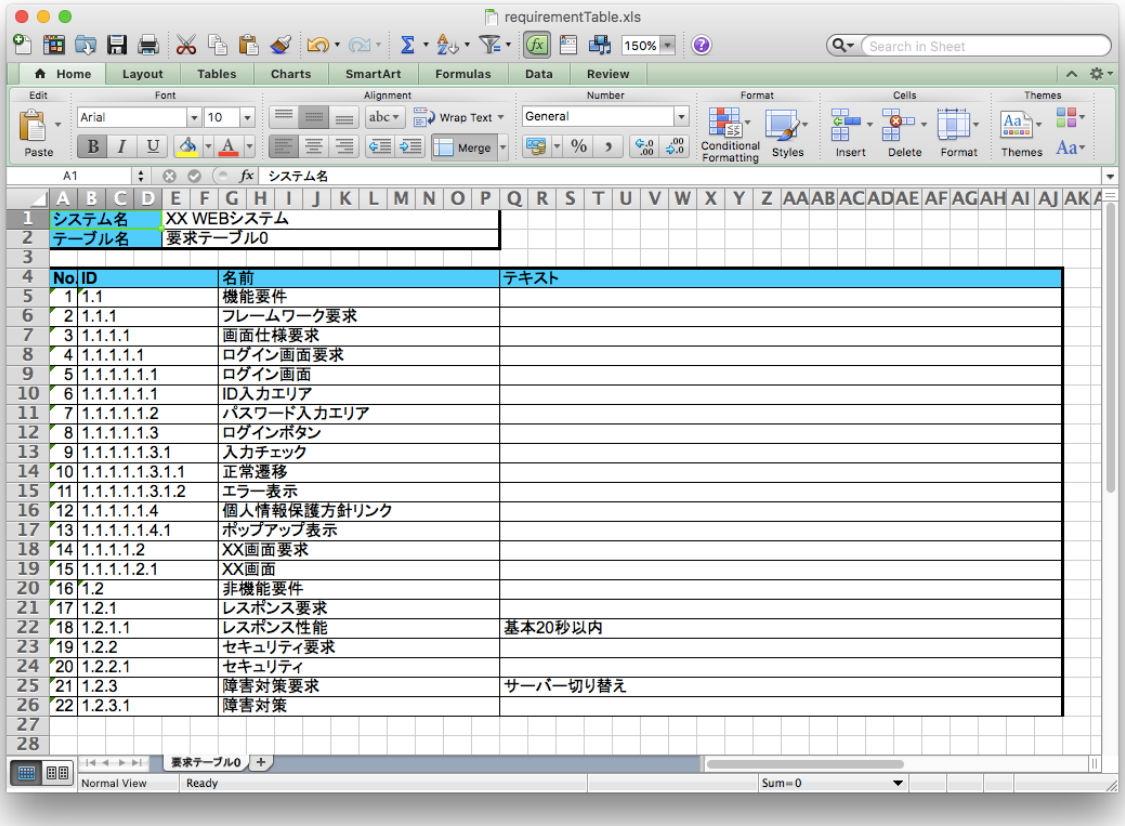
テストケースと要求間は、**検証**<<verify>>といって、要求を満たすことをテストケースで検証するための依存関係をはることができます。B さんは、プロジェクトマネージャーA さんが作ったモデルを、自分のモデルにマージし、検証関係を張っていきました。出来上がったモデルは、以下です。



その後、A さんは、要求開発リーダーの R さんに要求の詳細化を依頼しました。要求開発リーダーR さんは、要求テーブルを EXCEL で出力してチーム内の共有を開始しました。要求テーブルは、[ツール] – [要求] – [要求テーブルをExcel ファイルに出力]から出力できます。



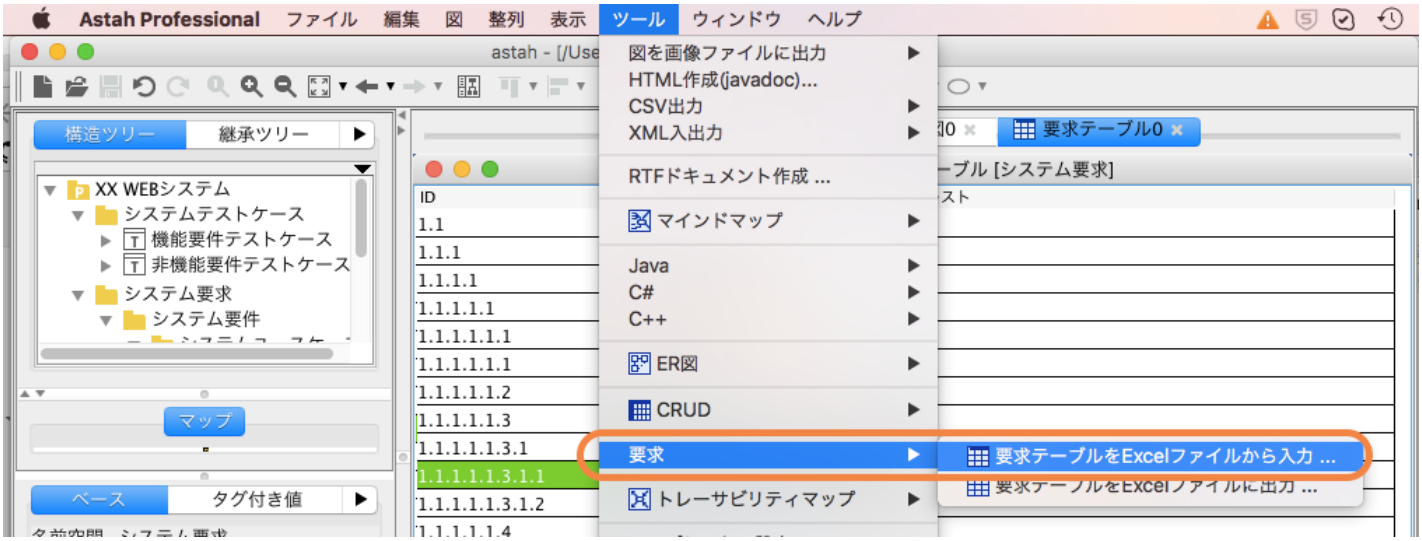
出力されたExcel は以下の通りです。



requirementTable.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
1	システム名	XX WEBシステム																																			
2	テーブル名	要求テーブル0																																			
3																																					
4	No	ID	名前	テキスト																																	
5	1	1.1	機能要件																																		
6	2	1.1.1	フレームワーク要求																																		
7	3	1.1.1.1	画面仕様要求																																		
8	4	1.1.1.1.1	ログイン画面要求																																		
9	5	1.1.1.1.1.1	ログイン画面																																		
10	6	1.1.1.1.1.1	ID入力エリア																																		
11	7	1.1.1.1.1.2	パスワード入力エリア																																		
12	8	1.1.1.1.1.3	ログインボタン																																		
13	9	1.1.1.1.1.3.1	入力チェック																																		
14	10	1.1.1.1.1.3.1.1	正常遷移																																		
15	11	1.1.1.1.1.3.1.2	エラー表示																																		
16	12	1.1.1.1.1.4	個人情報保護方針リンク																																		
17	13	1.1.1.1.1.4.1	ポップアップ表示																																		
18	14	1.1.1.1.2	XX画面要求																																		
19	15	1.1.1.1.2.1	XX画面																																		
20	16	1.2	非機能要件																																		
21	17	1.2.1	レスポンス要求																																		
22	18	1.2.1.1	レスポンス性能	基本20秒以内																																	
23	19	1.2.2	セキュリティ要求																																		
24	20	1.2.2.1	セキュリティ																																		
25	21	1.2.3	障害対策要求	サーバー切り替え																																	
26	22	1.2.3.1	障害対策																																		
27																																					
28																																					

要求テーブルを Excel 形式で受け取ったチームは、要求の階層化と要求テーブルの書式やルール決めを行い、要求を詰め、詳細化した Excel ファイルを、プロジェクトマネージャーA さんの.asta ファイルに反映させます。EXCEL 形式の要求テーブルは、[ツール] – [要求] – [要求テーブルをExcel ファイルに入力]から astah*に入力します。



Excel から要求テーブルを再入力する際、要求は、ID と名前を基準に更新されます。

要求が固まってきたところで、A さんは、要求開発リーダーR さんとレビューを繰り返し、要求を具体化していきました。その後、明確化された要求を元に、開発リーダーB さん中心に実装とテストケースを作成していきました。

[要求の機能のポイント]

SysML では、主に組み込み系を想定ターゲットとしていますが、それ以外のソフトウェア分野にも汎用的に利用できるものです。忘れられがちな非機能要件について、初期段階から強く意識できることなども要求モデリングの魅力の一つです。

また、astahで要求モデルを扱う利点は、要求間の関係を図示しやすいことと、要求とモデルの間の関係についても管理しやすくなることです。