

組込み分野のための UML モデル解説書



機能編 F002

自己診断

UMTP 組込み
モデリング部会

2016.5.15 更新

本書は、UML モデルカタログに含まれる「自己診断」のモデルの詳細を記述したものです。モデリングの初心者には教科書や参考書として、モデリングのベテランの方々にはモデルのヒントとして、ぜひともお手元に置いて活用してください。

UMTP は特定非営利活動法人 UML モデリング推進協議会の登録商標です。その他、本書に記載されている会社名、商品名などは、一般に各社の商標または登録商標です。

目 次

要求仕様	4
モデル一覧	11
エンティティに着目したモデル	12
分析モデル	13
PIM 設計モデル	25
メタファを使ったモデル	44
分析モデル	49
PIM 設計モデル	53

自動車や通信機器などには警告灯を備えるものや故障を示す表示ができるものがあります。これらには、装置が正しく動いているかどうかを調べる自己診断機能があり、異常を検知すると、(対処や)表示を行います。また、トラブル発生時に原因究明や対応を行うため、手動で自己診断ができる機器もあります。



要求仕様

1. 組込み装置には、装置が正しく動作しているかどうかを調べる自己診断機能があります。
2. 自己診断は、機器管理者の要求により起動される場合と、予め設定された条件に合わせて自動的に起動される場合があります。いずれも、機器管理者の指示により中止することが可能です。
3. 診断の対象には、デバイス単体や、複数デバイスから構成される機能ユニット、そして装置上で動作するアプリケーションが含まれます。
機能ユニットは階層構造を持つことが可能で、上位の機能ユニットは複数の下位機能ユニットを含むことができます。
4. 一つの装置内では複数の自己診断を実施することが可能です。各自己診断は、1つの診断対象に対する診断でも、それらを任意に組み合わせたものでも構いません。
(例：起動時診断や月次診断では全機能ユニット、週次診断では重要機能ユニット、日次診断では最重要ユニットなど。)
また、自己診断に含まれる各診断は、並行・逐次、どちらの形でも実施可能です。
5. 診断対象ごとに、診断項目が決まっており、各診断ではどの診断項目を診断するかを設定することが可能です。
(例：起動時診断や月次診断では診断項目 1～10 まで、週次診断では 1～5 と 6～10 を交互に、日次診断では 1 と 2 のみなど。)
6. 診断結果は、診断項目ごとに OK/NG の結果と、NG の場合には、エラーコードをレポートします。
7. 診断結果の履歴は保存する必要がありません。
8. 診断途中で中止されたときには、それまでの診断内容をレポートして終了します。

ユースケース

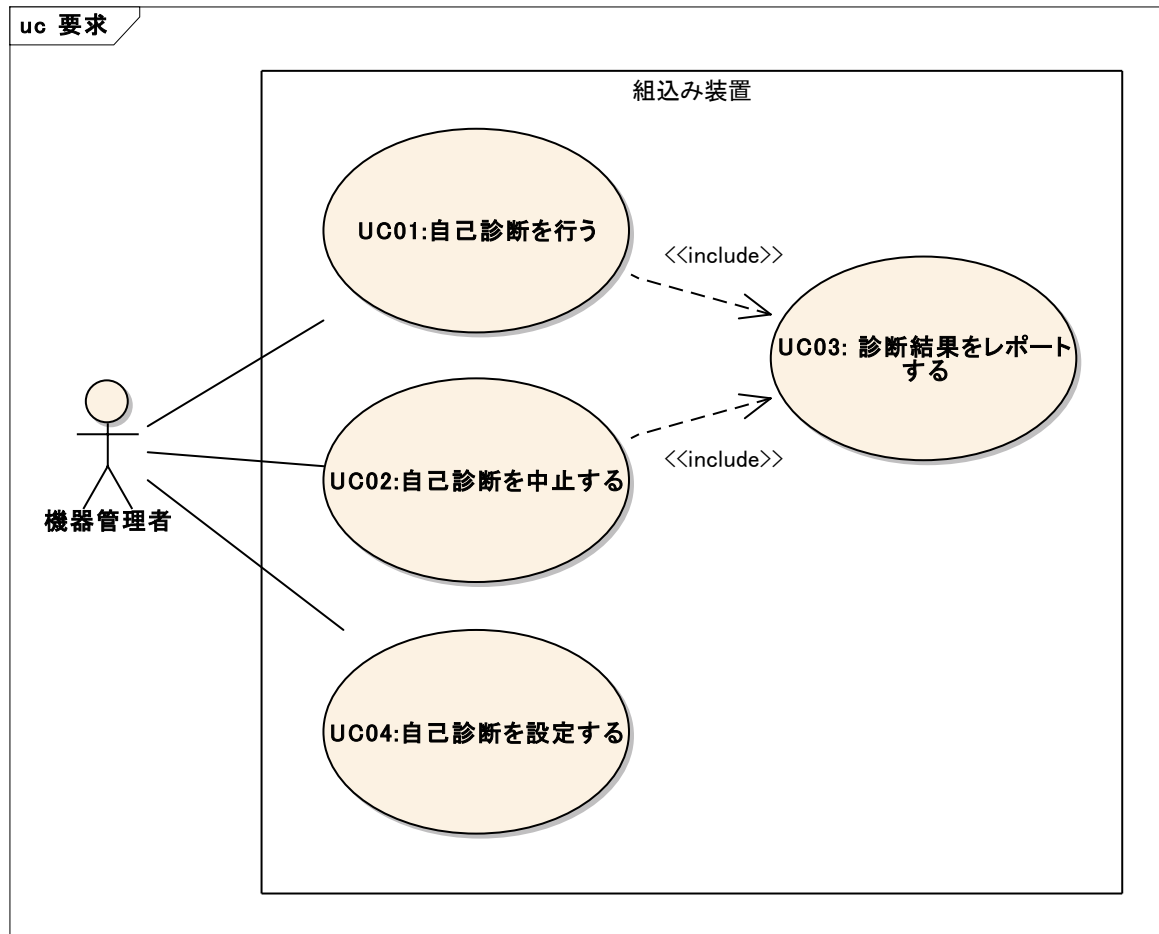


図1

ユースケース記述

<UC01:自己診断を行う>

■ 概要

機器管理者が組込み装置に診断開始を指示すると、組込み装置は診断を行い、診断結果をレポートする。

■ アクター

機器管理者

■ 事前条件

- ・ 組込み装置に実行する自己診断が設定されている

■ 事後条件

- ・ 自己診断が終了し、診断結果がレポートされている

■ メインフロー

1. アクターは、実行する自己診断を指定し、システムに診断開始を指示する
2. システムは、診断を行う
3. システムは、「診断結果をレポートする」ユースケースを実行する
4. UC を終了する

■ 課題や T. B. D 項目

なし

■ 備考

- ・ 複数の自己診断を並行して行うことが可能。この際、組込み装置内で必要な排他制御を行う。

<UC02:自己診断を中止する>

■概要

組込み装置が自己診断実行中に、機器管理者が自己診断中止を指示すると、組込み装置は自己診断を中止し、それまでの診断結果をレポートする。

■アクター

機器管理者

■事前条件

- ・組込み装置は自己診断を実行中である

■事後条件

- ・自己診断が中止された
- ・診断結果がレポートされた

■メインフロー

1. アクターは、システムに、自己診断の中止を指示する
2. システムは、自己診断を中止する
3. システムは、「診断結果をレポートする」ユースケースを実行する
4. UC を終了する

■課題や T. B. D 項目

- ・複数の自己診断を並行して実行中の場合、中止する診断の指定方法(またはすべてを中止)

■備考

- ・自己診断を実行していない場合に、本ユースケースを実行しても、何も行われない(エラーにならない)。自己診断中止の指示中に自己診断を終了した場合も同様。

<UC03: 診断結果をレポートする>

■ 概要

組込み装置が診断結果をレポートする。

■ アクター

なし

■ 事前条件

なし

■ 事後条件

診断結果がレポートされている

■ メインフロー

1. システムは、診断結果をレポートする
2. UC を終了する

■ 課題や T. B. D 項目

- ・ レポートの出力手段
- ・ レポートのフォーマット(形式)

■ 備考

- ・ 本ユースケースは他のユースケースから起動される

<UC04:自己診断を設定する>

■概要

機器管理者が組込み装置に自己診断を設定する。

■アクター

機器管理者

■事前条件

なし

■事後条件

組込み装置に自己診断が設定されている

■メインフロー

1. アクターは、システムに、自己診断を設定することを指示する
2. アクターは、システムに、各種設定内容を指示する
3. システムは、指示された内容の自己診断を保存する
4. UC を終了する

■代替フロー

なし

■課題や T. B. D 項目

- ・ どの自己診断を並行・逐次で実行するか の指定方法
- ・ レポートの出力手段とレポートのフォーマット(形式) の指定方法

■備考

設定内容は次の通り。

- ・ 自己診断の名前
- ・ 自己診断の起動方法
- ・ 診断の種類（1 つの診断対象に対する診断 or 既存の自己診断を組み合わせた診断）
 - 1 つの診断対象に対する診断を設定した場合は、1 つの診断対象を指定する
 - 既存の自己診断を組み合わせた診断の場合は、複数の自己診断を指定する

シナリオ

<SC0101:自己診断を行う>

1. 山田太郎さんがルーターにイーサネット診断の開始を指示する。
2. ルーターは診断を行う。
3. ルーターは診断結果をレポートする。

<SC0201: 診断を中止する>

1. 山田太郎さんがルーターにイーサネット診断の開始を指示する。
2. ルーターは診断を行う。
3. 山田太郎さんがルーターに診断の中止を指示する。
4. ルーターは自己診断を中止する。
5. ルーターは診断結果をレポートする。

<SC0401: 自己診断を設定する(1 つの診断対象に対する診断)>

1. 山田太郎さんがルーターに自己診断を設定することを指示する。
2. 山田太郎さんはルーターに設定する自己診断の名前を『イーサネットメモリチェック』と指示する。
3. 山田太郎さんはルーターに自己診断の起動方法を毎日午前 1 時と指示する。
4. 山田太郎さんはルーターに「1 つの診断対象に対する診断」を設定することを指示する。
5. 山田太郎さんはルーターに自己診断の診断対象をイーサネットと指示する。
6. 山田太郎さんはルーターに自己診断の診断内容をメモリチェックと指示する。
7. ルーターは毎日午前 1 時、イーサネットを対象に、メモリチェックの自己診断を行うように設定する。

<SC0402: 自己診断を設定する(1 つの診断対象に対する診断その 2)>

1. 山田太郎さんがルーターに自己診断を設定することを指示する。
2. 山田太郎さんはルーターに設定する自己診断の名前を『イーサネットレジスタチェック』と指示する。
3. 山田太郎さんはルーターに自己診断の起動方法を毎日午前 2 時と指示する。
4. 山田太郎さんはルーターに「1 つの診断対象に対する診断」を設定することを指示する。
5. 山田太郎さんはルーターに自己診断の診断対象をイーサネットと指示する。
6. 山田太郎さんはルーターに自己診断の診断内容をレジスタチェックと指示する。
7. ルーターは毎日午前 2 時、イーサネットを対象に、レジスタチェックの自己診断を行うように設定する。

<SC0403: 自己診断を設定する(既存の自己診断を組み合わせた診断)>

1. 山田太郎さんがルーターに自己診断を設定することを指示する。
2. 山田太郎さんはルーターに設定する自己診断の名前を『イーサネット』と指示する。
3. 山田太郎さんはルーターに自己診断の起動方法を毎週日曜日午前 3 時と指示する。
4. 山田太郎さんはルーターに「既存の自己診断を組み合わせた診断」を設定することを指示する。
5. 山田太郎さんはルーターに『イーサネットメモリチェック』自己診断を組み合わせるように指示する。
6. 山田太郎さんはルーターに『イーサネットレジスタチェック』自己診断を組み合わせるように指示する。
7. ルーターは毎週日曜日午前 3 時、『イーサネットメモリチェック』自己診断と『イーサネットレジスタチェック』自己診断を組み合わせた自己診断を行うように設定する。

モデル一覧

着目点	コンセプト	ポイント
機能に着目したモデル	なし	
エンティティに着目したモデル	要求仕様の内容をエンティティを中心に分析し、診断と診断項目の静的構造に着目したモデルです。	要求仕様の内容を忠実に分析しています。
状態に着目したモデル	なし	
メタファを使ったモデル	装置の自己診断機能を、日常の診断(健康診断など)にたとえてモデリングしています。	身近な話題から入るの で、モデリングの導入編 としてお勧めです。

エンティティに着目したモデル

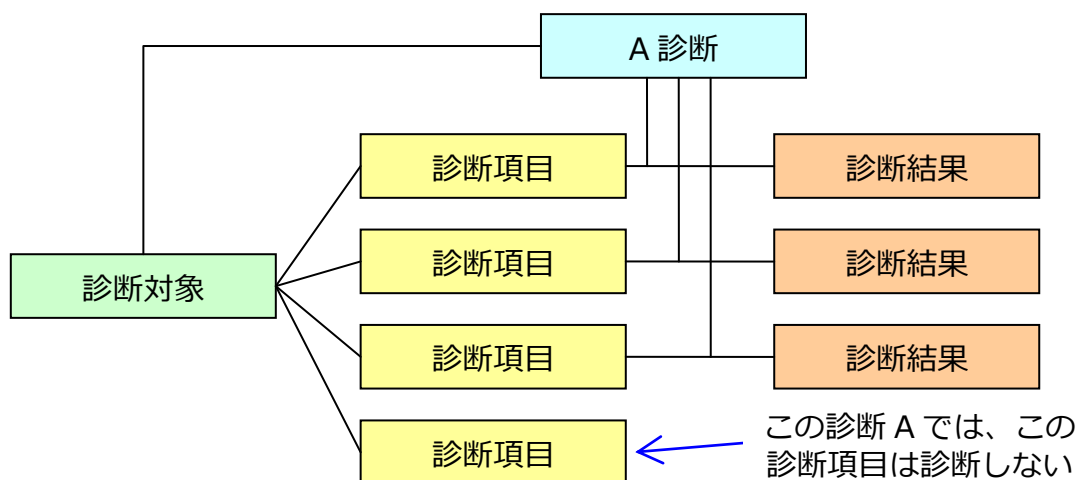
要求仕様をエンティティに着目し、診断を診断対象の診断項目と対応する結果と捉え、このような診断を組み合わせることでさまざまな診断をモデリングしています。

モデリングのコンセプト

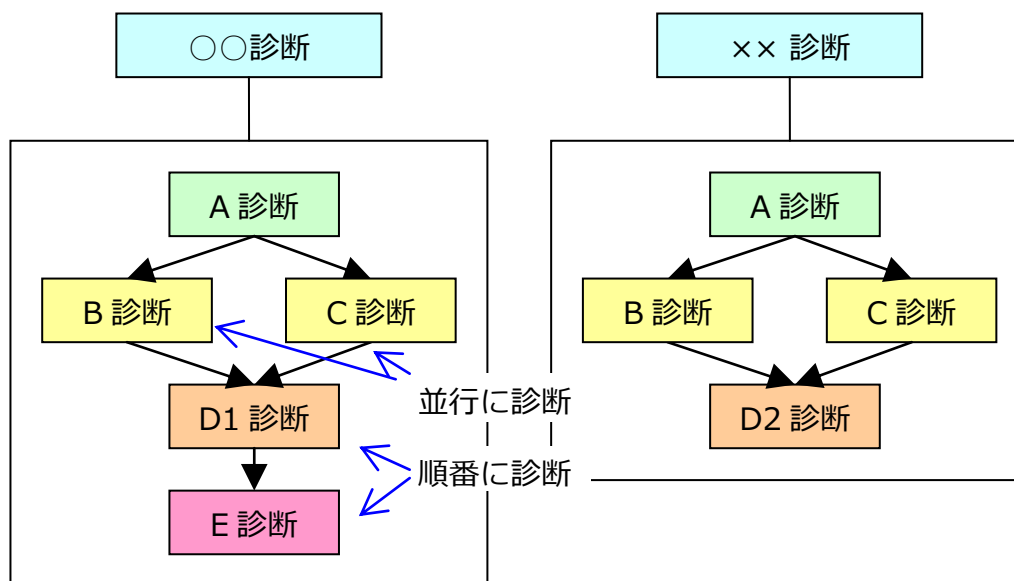
要求仕様の内容を忠実に分析し、モデリングします。

要求仕様から以下の特徴が読み取れました。

- (i) (後述の組み合わせた診断に対して) 単一の対象に対する診断とは 1 個の診断対象に対して定められた診断項目を診断し、診断結果を得ることです。診断対象に対して診断項目は決まっていますが、診断毎にどの診断項目を診断するかを設定することが可能です。(UML 図ではなく)簡単に図示すると以下ようになります。



- (ii) 診断は複数の診断から組み合わせることもでき、以下のように 並行/順番に実行されます(組み合わせた診断)。(UML 図ではなく)簡単に図示すると以下ようになります。



診断条件、診断対象、診断項目、レポートは関連がない場合もあります。組み合わせた診断の場合、個々の診断には必要ないので、関連がない場合もあります。また、機器管理者の要求により起動される場合には、診断条件がない場合もあります。

診断結果は診断と診断項目に関連があります。どの診断のどの診断項目の診断結果なのかを表します。診断を行っていない場合、診断結果はありません。

組み合わせた診断は診断の自己関連で表されます。実施する診断を順番に繋ぎます。この際に 1 個の診断から複数の診断へ分岐すると、それらの複数の診断が並列に実施されることを表します。また、複数の診断から 1 個の診断に合流すると、並列の実施から順次の実施となります。

組込み装置は機能ユニットから構成され、機能ユニットはデバイスから構成されます。機能ユニットは構造をとることも可能で、ある機能ユニットは機能ユニットから構成されることもあります。また、アプリケーションも組込み装置の構成要素です。

デバイス、機能ユニット、アプリケーションは診断対象です。すなわち、これらを汎化したものが診断対象です。

レポートは、出力方法とフォーマット(形式)があります。

出力方法の例としてはコンソール出力、(ログ)ファイル、電子メール、Web ページ、LED などがあります。

フォーマットの例としてはプレーンテキスト、MIME、HTML などがあります。

レポート、出力方法、フォーマットは意味的にまとまっているので、レポートドメインとします。

組込み装置(ルーター)のオブジェクト構成

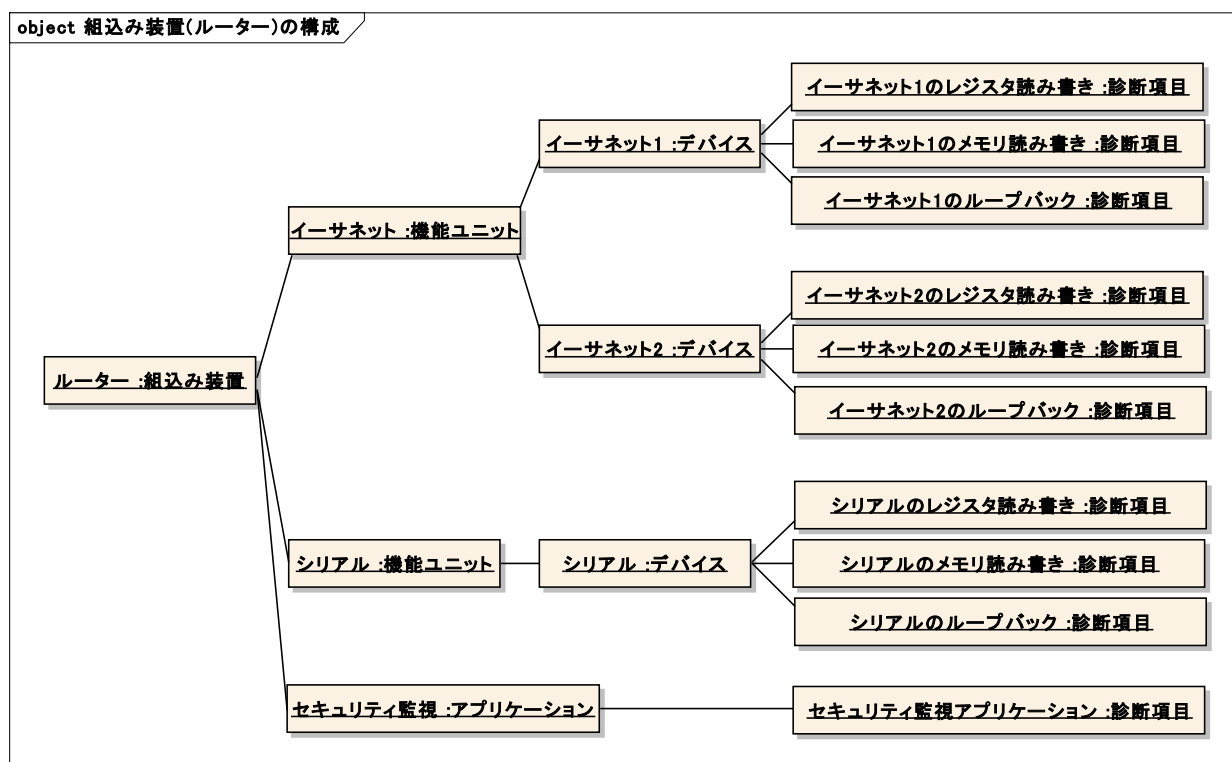


図3

イーサネット×2、シリアル×1 のルーターのオブジェクト構成です。

「診断、診断対象、診断項目、診断結果の関係」のオブジェクト構成

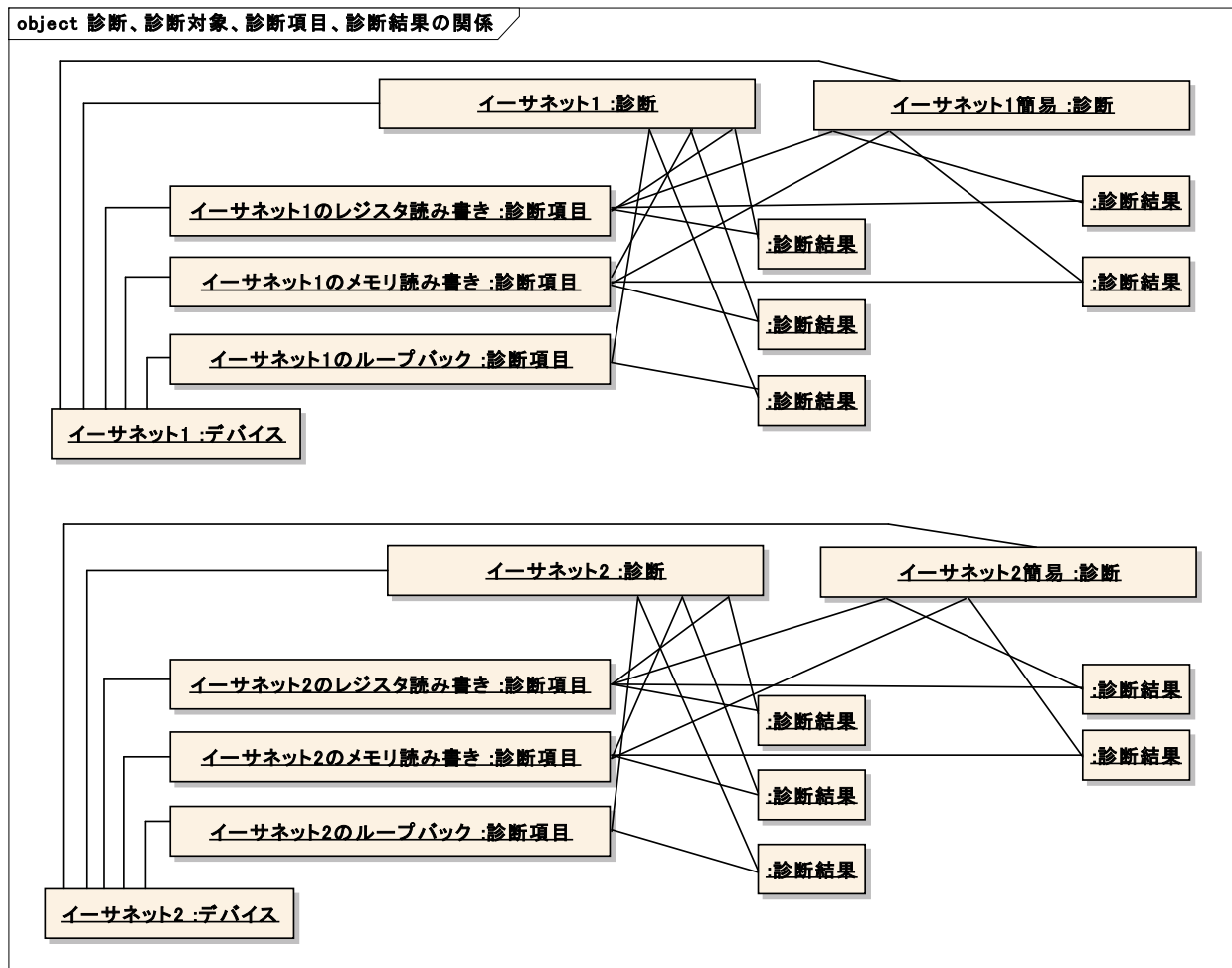


図4

先のルーターで診断と診断に関連するモノを示したオブジェクト構成です。

「イーサネット:デバイス」は「レジスタ読み書き:診断項目」、「メモリ読み書き:診断項目」、「ループバック:診断項目」と関連があり、デバイスのインスタンスが異なると、同じ診断項目でも別の診断項目のインスタンスとなります。

ある診断対象(「イーサネット:デバイス」)に対する診断は、診断対象のすべての診断項目を行ってもよいですし(「イーサネット 1: 診断」や「イーサネット 2: 診断」)、一部の診断項目を行ってもよいです(「イーサネット 1 簡易: 診断」や「イーサネット 2 簡易: 診断」)。

「組み合わせた診断」のオブジェクト構成

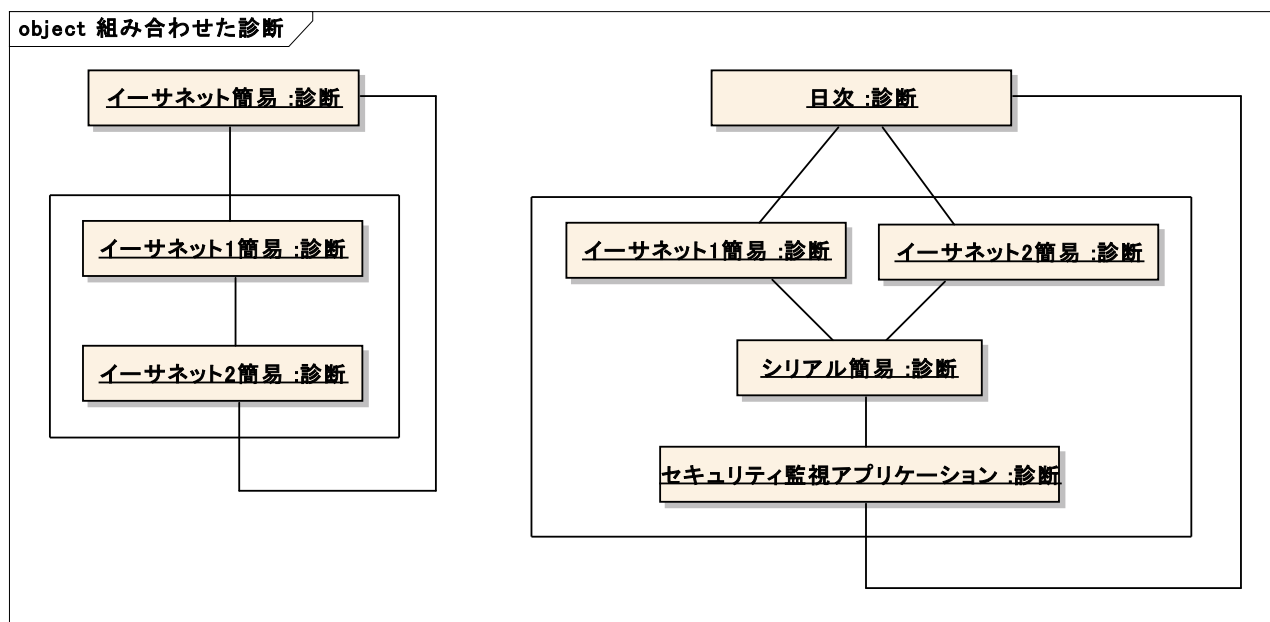


図5

同じく先のルーターで組み合わせた診断のオブジェクト構成です。

「イーサネット簡易:診断」は「イーサネット 1 簡易:診断」を実施し、次に「イーサネット 2 簡易:診断」を実施します。

「日次:診断」は「イーサネット 1 簡易:診断」と「イーサネット 2 簡易:診断」を並列に実施し、次に「シリアル簡易:診断」を実施し、次に「セキュリティ監視アプリケーション:診断」を実施します。

「イーサネット 2 簡易:診断」から「イーサネット簡易:診断」へのリンク、「セキュリティ監視アプリケーション:診断」から「日次:診断」へのリンクは診断の終了を通知するためにあります。



動的モデル

操作の検討と実現性の確認のため、ユースケースのメインフローまたはシナリオからシーケンス図を作成しました。オブジェクト図と同様にルーターを題材としています。

ユースケース「UC01:自己診断を行う」の相互作用

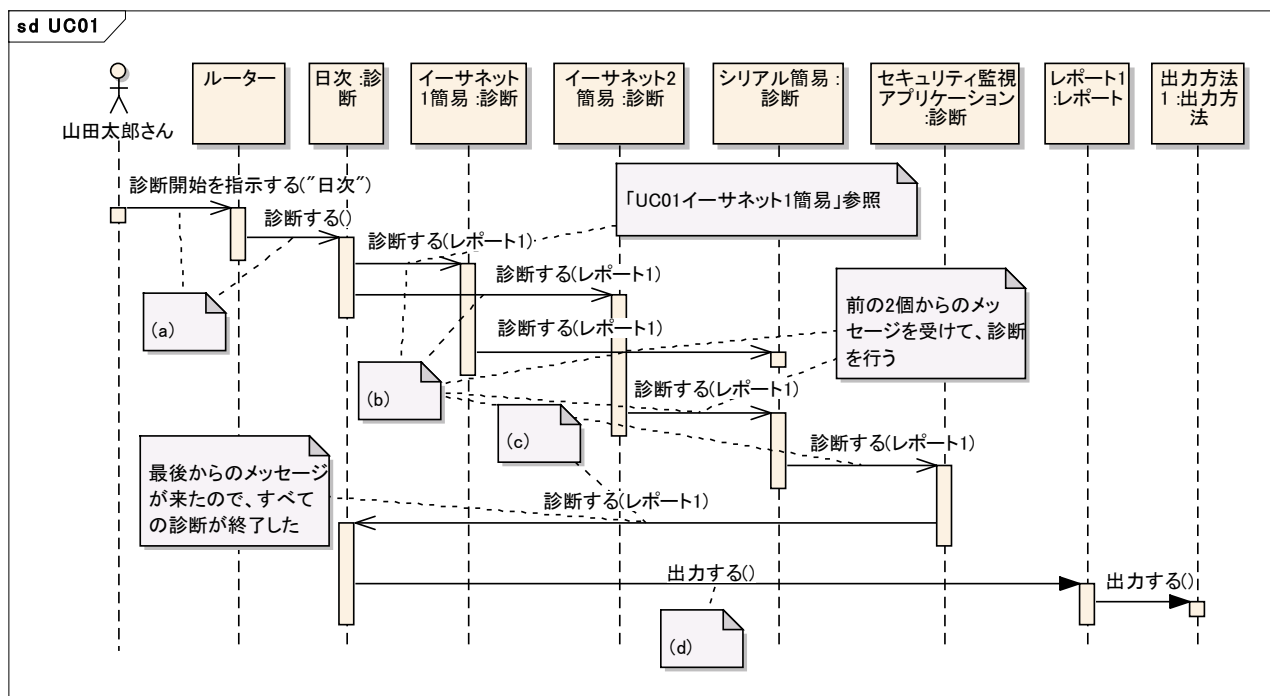


図7

- (a)アクターから診断開始の指示を受けると、ルーターは該当する診断インスタンスに「診断する」メッセージを送信します。
- (b)「日次:診断」から診断の構造に従って、個々の診断へ「診断する」メッセージを送信します(「日次:診断」から「イーサネット1簡易:診断」への「診断する」メッセージなど)。
- (c)診断の構造の最後にある「セキュリティ監視アプリケーション:診断」が完了を示す「診断する」メッセージを「日次:診断」へ送信します。
- (d)「日次:診断」がレポートに出力します。

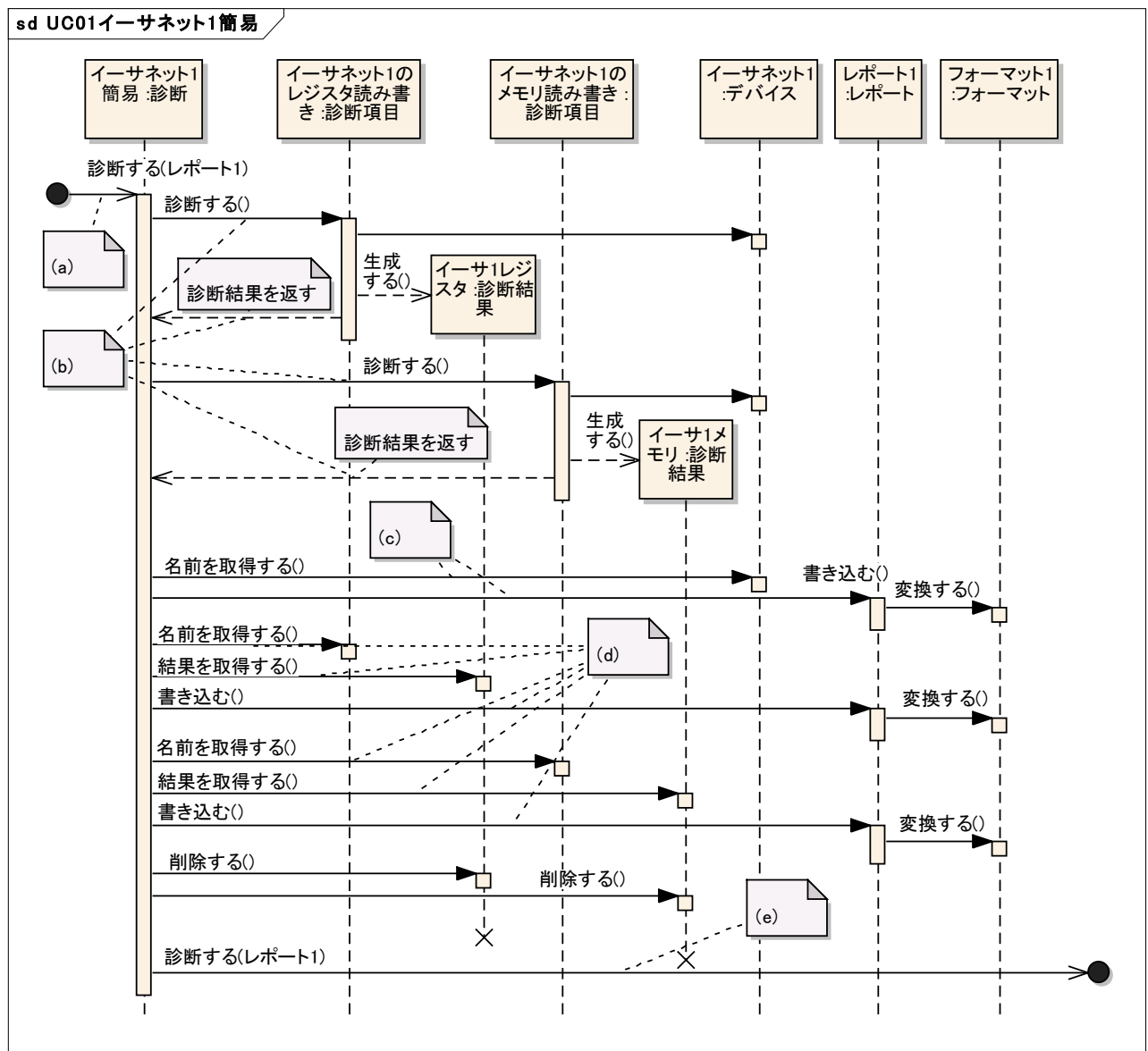
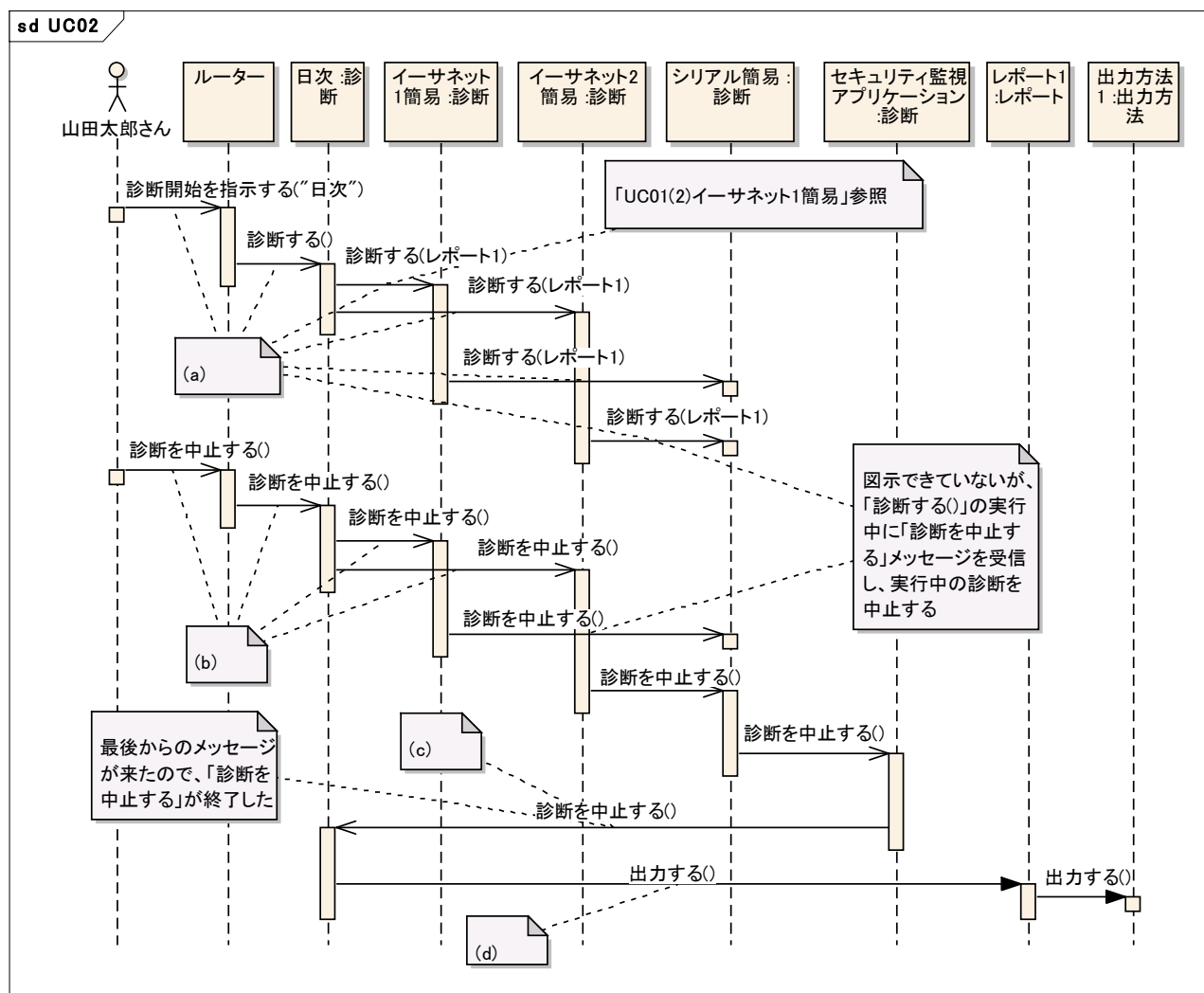


図8

- (a)「イーサネット 1 簡易:診断」は「診断する」メッセージを受信します。
- (b)「イーサネット 1 簡易:診断」は各診断項目へ「診断する」メッセージを送信し、診断結果を受け取ります。
- (c)「イーサネット 1 簡易:診断」はデバイスから名前を取得し、レポートに書き込みます。
- (d)「イーサネット 1 簡易:診断」は各診断結果から結果を取得し、レポートに書き込みます。
- (e)(すべて終了したので)「イーサネット 1 簡易:診断」は「次」の(関連がある)「診断」へ「診断する」メッセージを送信します。

ユースケース「UC02:自己診断を中止する」の相互作用



- (a)アクターから診断開始の指示を受けると、ルーターは「ユースケース「UC01:自己診断を行う」のシーケンス」と同様に動きます。
- (b)アクターから診断の中止の指示を受けると、「日次:診断」から診断の構造に従って、個々の診断へ「診断を中止する」メッセージを送信します(「日次:診断」から「イーサネット1簡易:診断」への「診断を中止する」メッセージなど)。
- (c)診断の構造の最後にある「セキュリティ監視アプリケーション:診断」が完了を示す「診断を中止する」メッセージを「日次:診断」へ送信します。
- (d)「日次:診断」がレポートに出力します。

シナリオ「SC0401: 自己診断を設定する(1つの診断対象に対する診断)」の相互作用

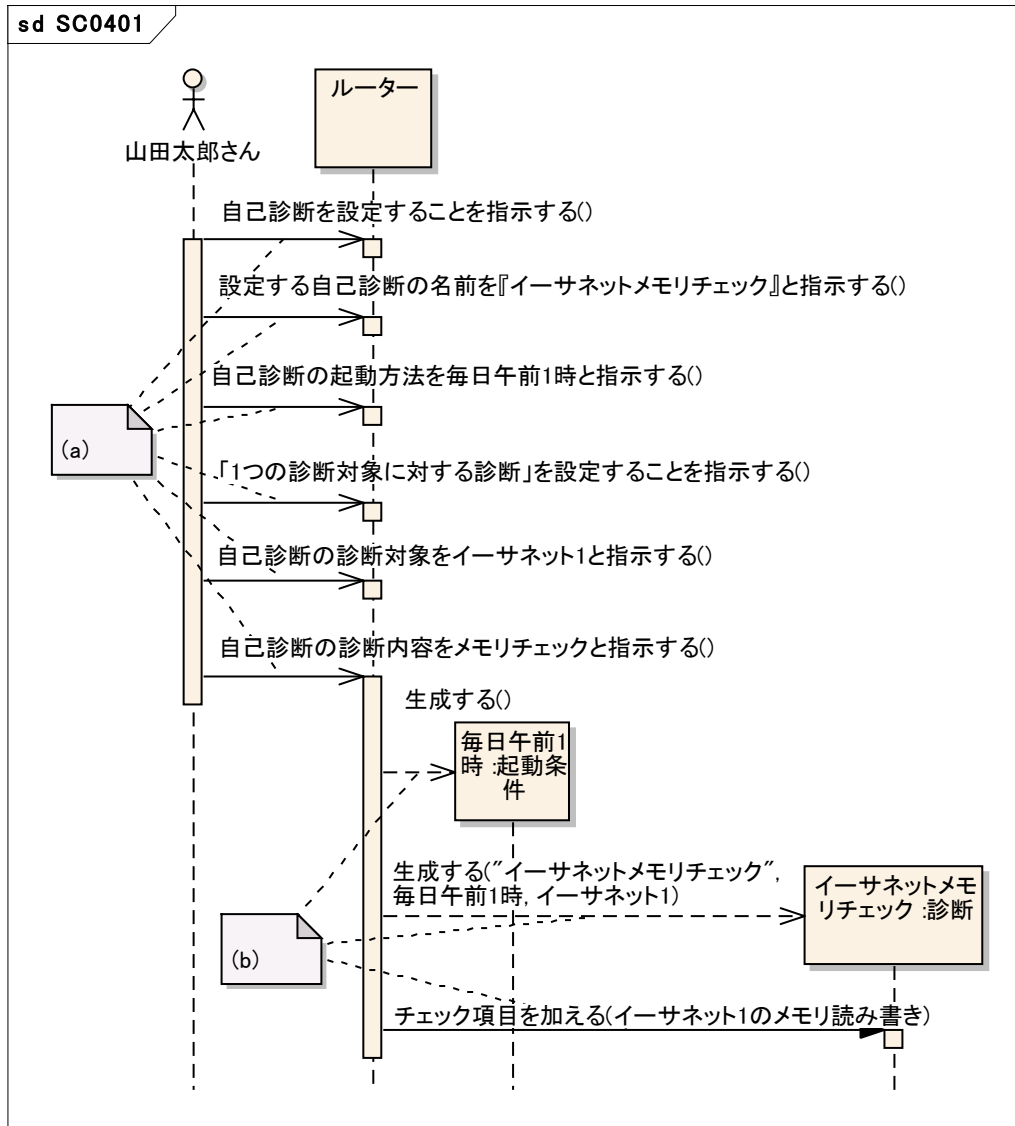


図10

(a)アクターはルーターへ自己診断の名前、起動方法、診断対象、診断内容を指示します。

(b)ルーターは起動条件と診断を生成し、関連付けます。

シナリオ「SC0403: 自己診断を設定する(既存の自己診断を組み合わせた診断)」の相互作用

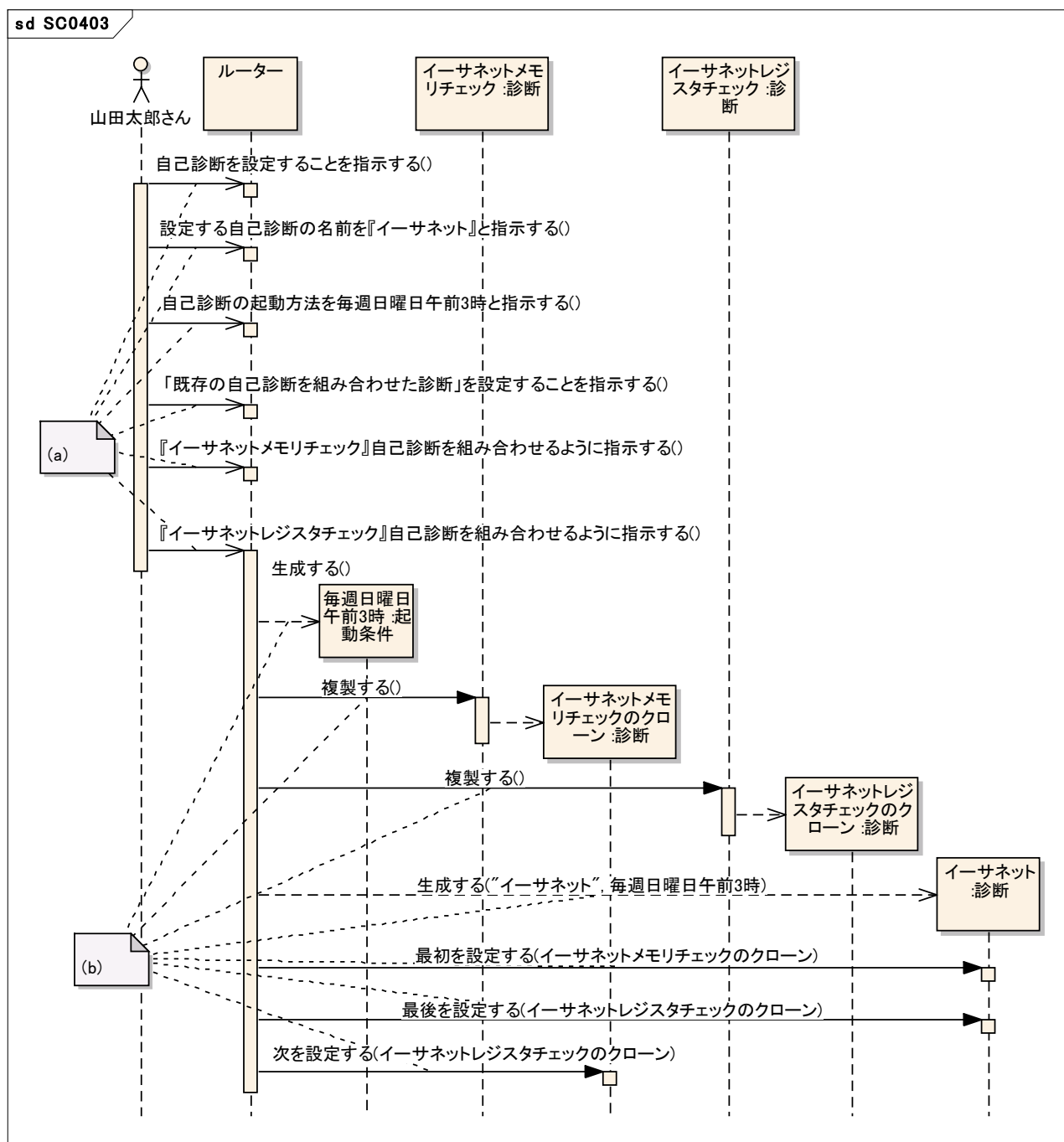


図11

(a)アクターはルーターへ自己診断の名前、起動方法、組み合わせる診断を指示します。

(b)ルーターは起動条件と診断を生成し、診断を複製し、関連付けます。

初期化時の相互作用

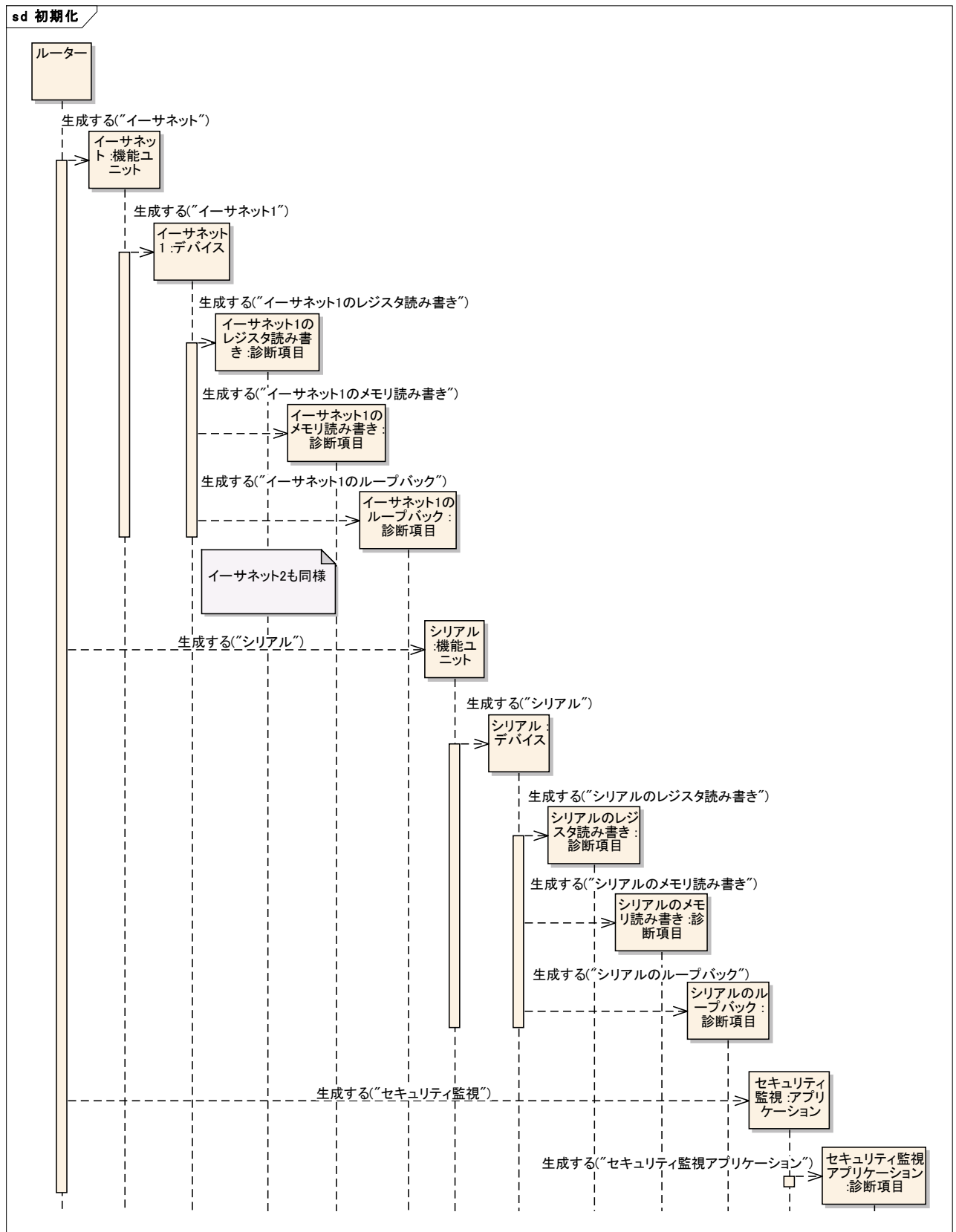


図12

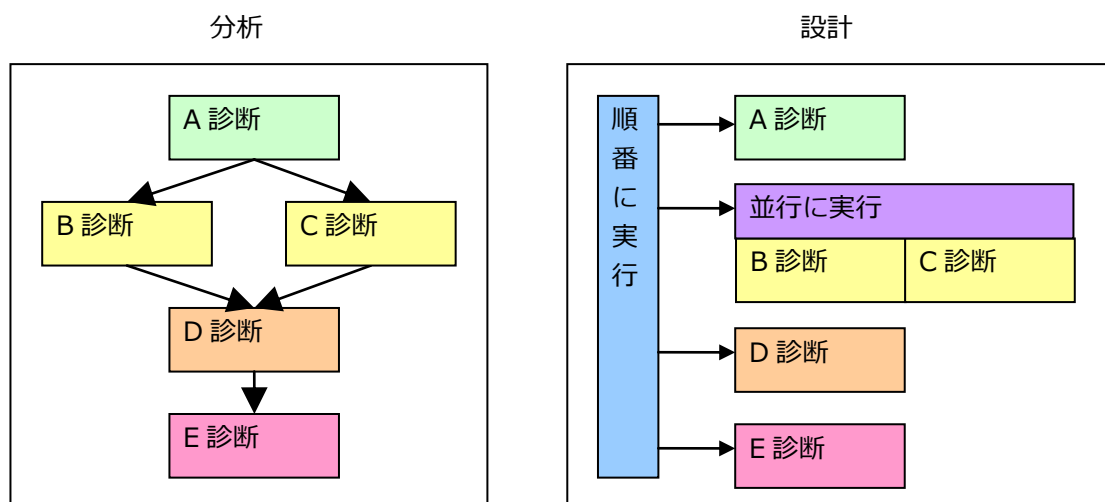
ユースケースにはありませんが、初期化の際のシーケンスです。

ルーター(組込み装置)は機能ユニットとアプリケーションを生成し、機能ユニットはデバイスを生成し、デバイスとアプリケーションは診断項目を生成します。

PIM 設計モデル

実装できるように、かつプラットフォームに依存しないように、分析クラスを詳細化します。具体的には以下の通りです。

- (i)分析では診断の順番は診断インスタンスのネットワークで表されていました。このように実装すると、流れが分岐し、再度合流する処理や、終了を通知する処理の実現が煩雑になりそうです。このため、設計では、診断を順番に行うものと並行して行うものを導入します。



- (ii)並列に実行するために、スレッドを導入します。ただし、プラットフォームに依存しないように、汎用的なライブラリとして導入し、PSM で実装することにします。
- (iii)スレッドを利用して、診断を実行する「診断実行」クラスを導入します。
- (iv)自己診断を中止するため、実行する/した各診断にユニークな ID を付けます。これにより、自己診断を中止している最中に自己診断を完了してしまうなどのきわどいタイミングでも動作が不安定にならないようにします。
- (v)各インスタンスを管理する「～管理」クラスを導入します。
- (vi)誘導可能性の明示など、設計モデルとして詳細化します。

静的モデル

ドメイン構成

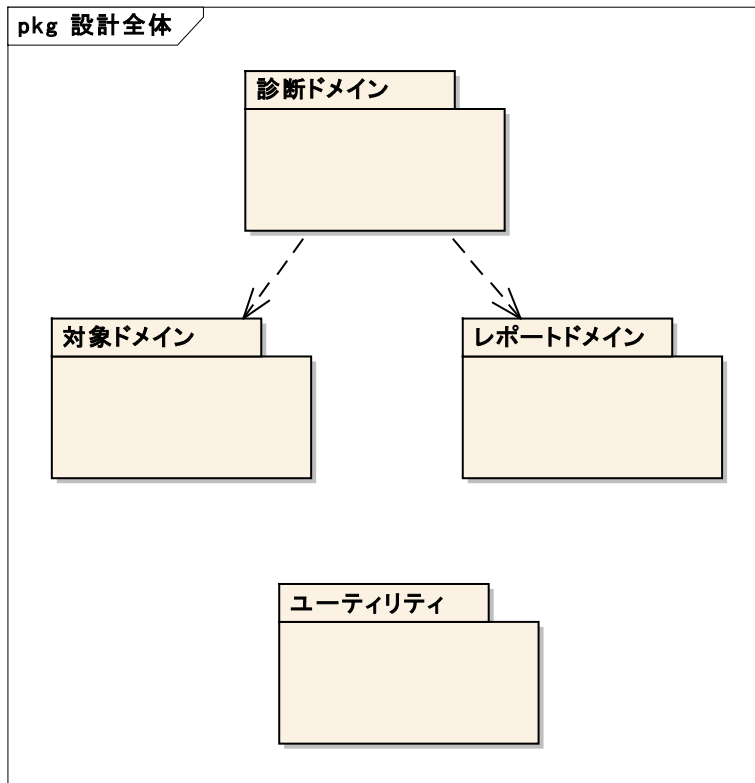


図13

設計ではクラスが非常に増えたため、3 個のドメインとユーティリティに分けました。

診断ドメイン：自己診断の中心となる「診断」クラスを中心とする部分

対象ドメイン：「診断対象」クラスを中心とする部分

レポートドメイン：「レポート」クラスを中心とする部分

ユーティリティ：スレッドなどを実現する汎用的なライブラリ

「診断ドメイン (1)診断関連」のクラス構成

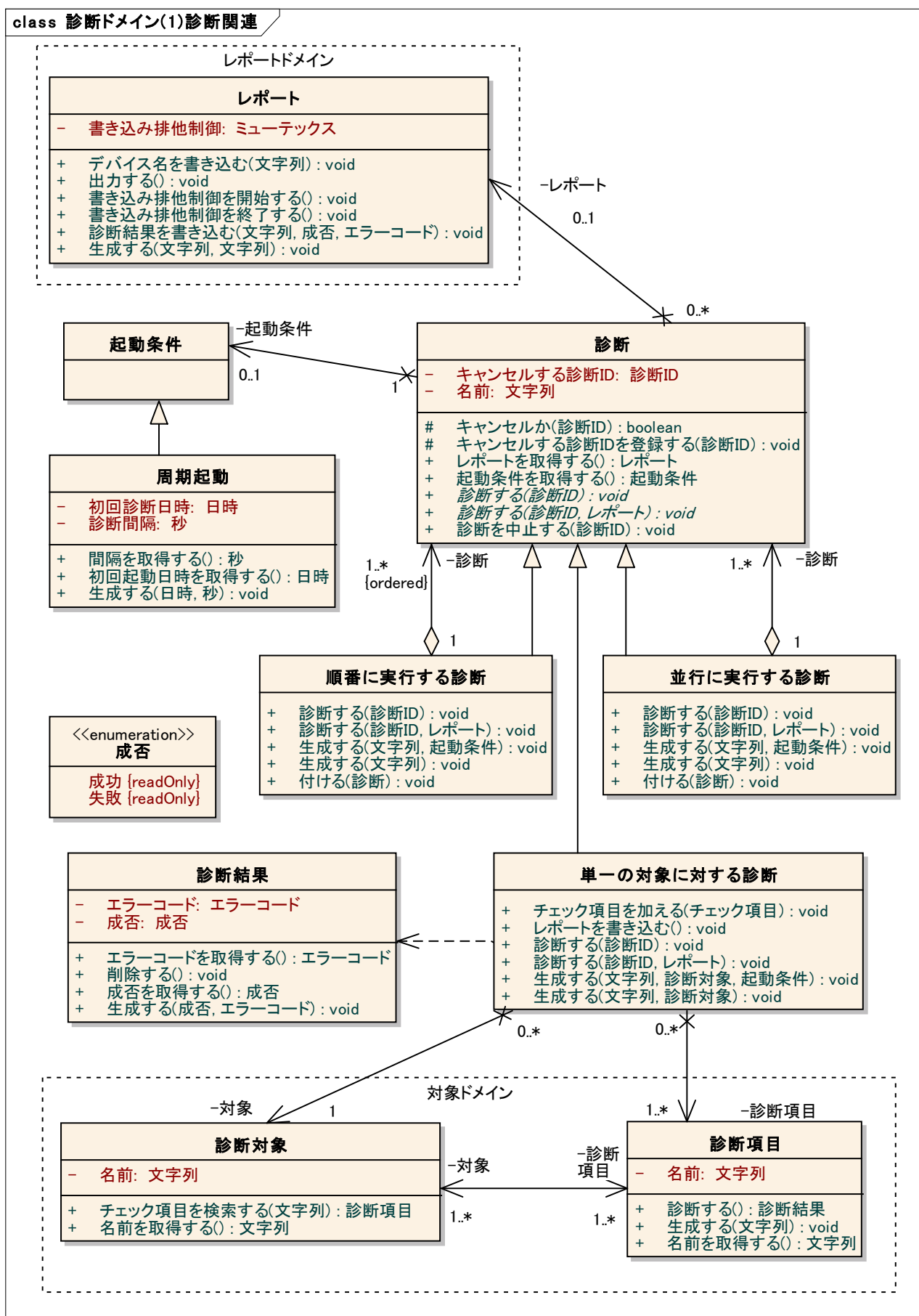


図14

スレッドを作成する/しないが異なるため、診断クラスを「順番に実行する診断」クラスと「並行に実行する診断」クラスに分けました。また、「単一の対象に対する診断」も振る舞いが異なるので、異なるクラスとしました。

「単一の対象に対する診断」は文字通り単一の対象に対して、指定された診断項目の診断を実行します。

「並行に実行する診断」はスレッドを作成し、指定された診断を並行に実行します。

「順番に実行する診断」はスレッドを作成せず、指定された診断を順番に実行します。

「診断結果」と「診断」の関連は「診断する」操作の間だけ必要なので、依存としました。

「診断ドメイン (2)診断管理関連」のクラス構成

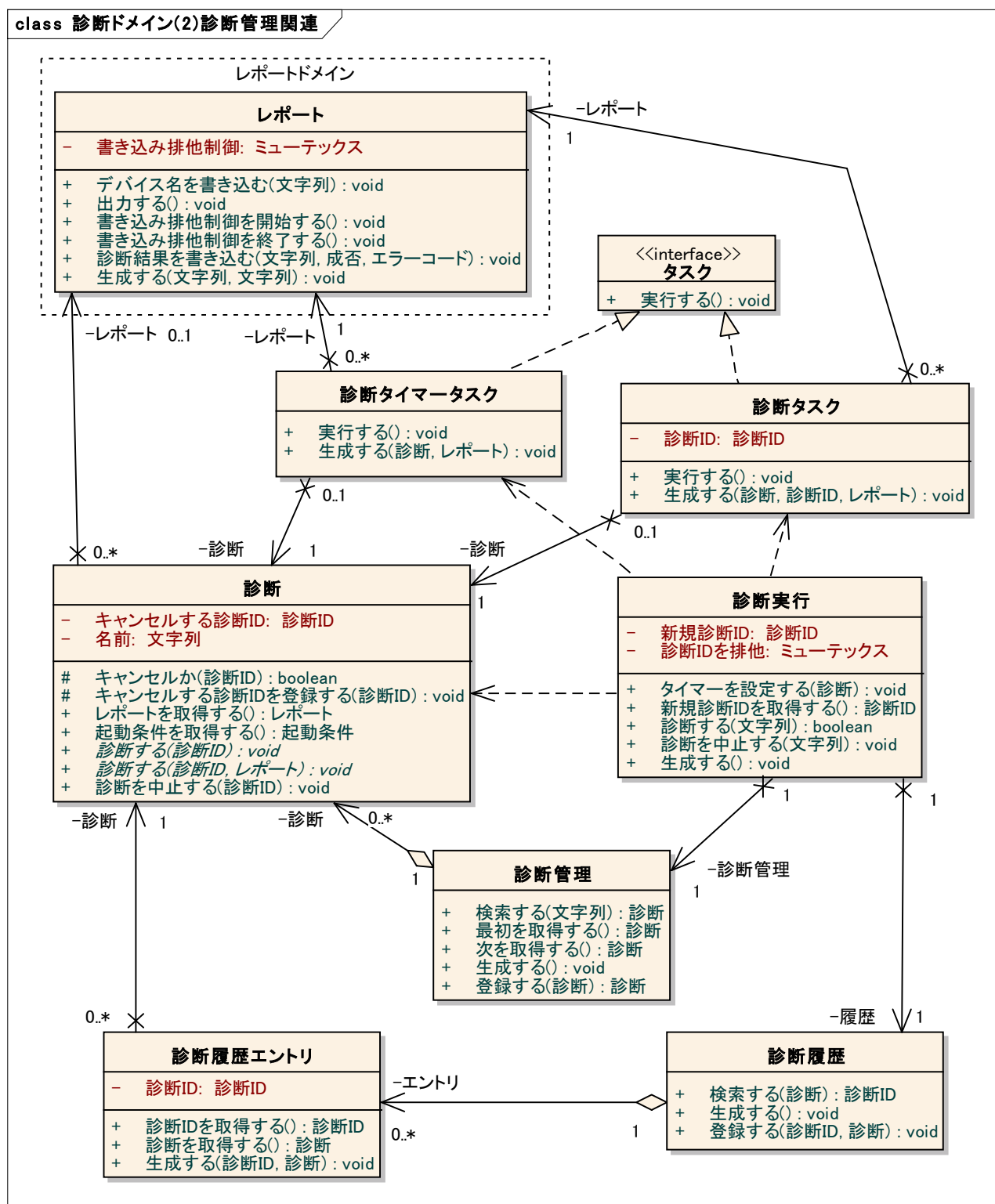


図15

診断管理は診断を集約し、管理します。

診断実行は診断の実行を司ります。タスクインタフェースを実装した診断タスクと診断タイマータスクを利用し、診断を実行します。

診断履歴は診断を行う際に診断履歴エントリを作成し、診断の履歴を管理します。履歴は自己診断を中止する際に使用します。

「対象ドメイン」のクラス構成

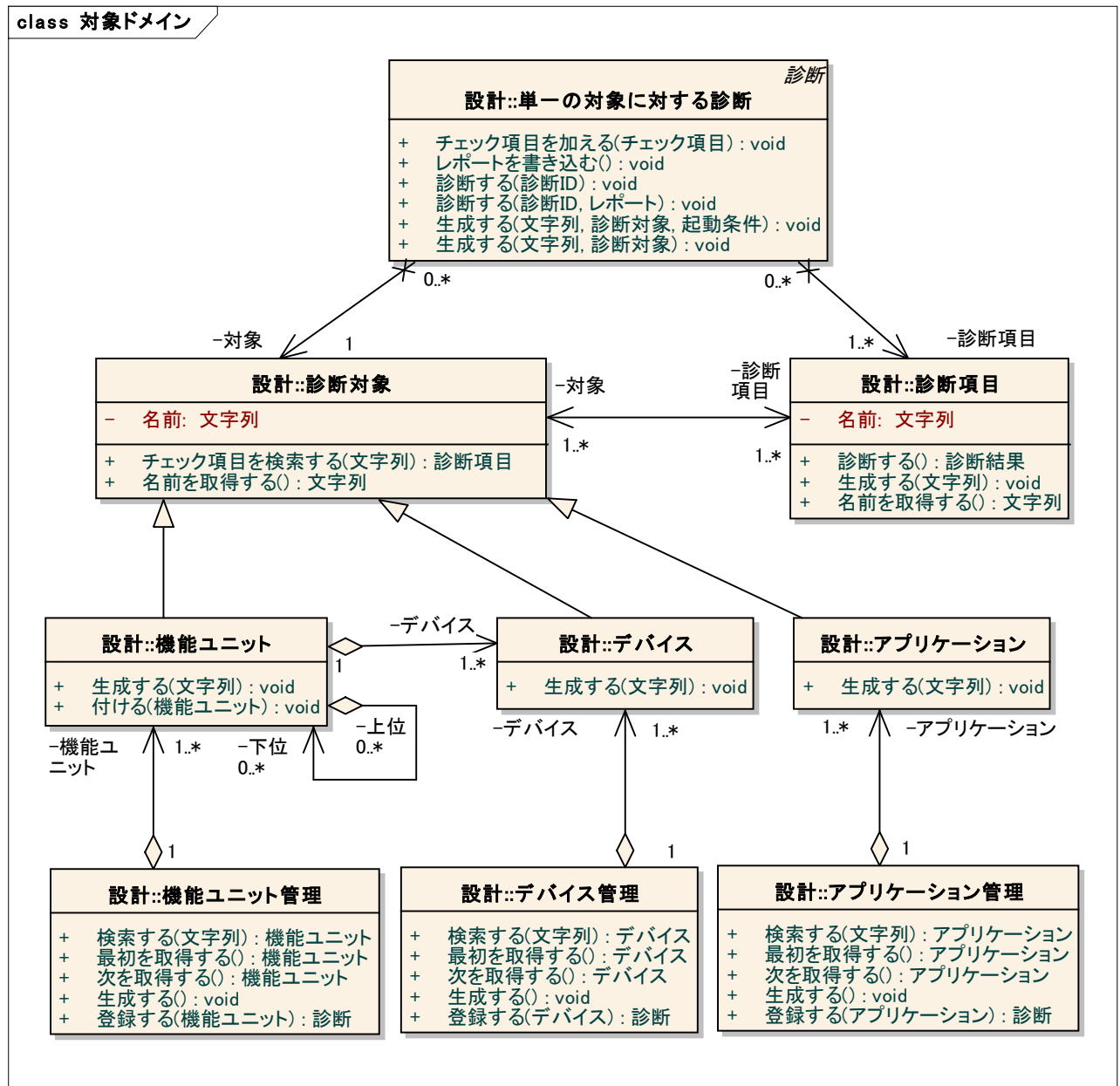


図16

機能ユニット管理は機能ユニットを集約し、管理します。

デバイス管理はデバイスを集約し、管理します。

アプリケーション管理はアプリケーションを集約し、管理します。

「レポートドメイン」のクラス構成

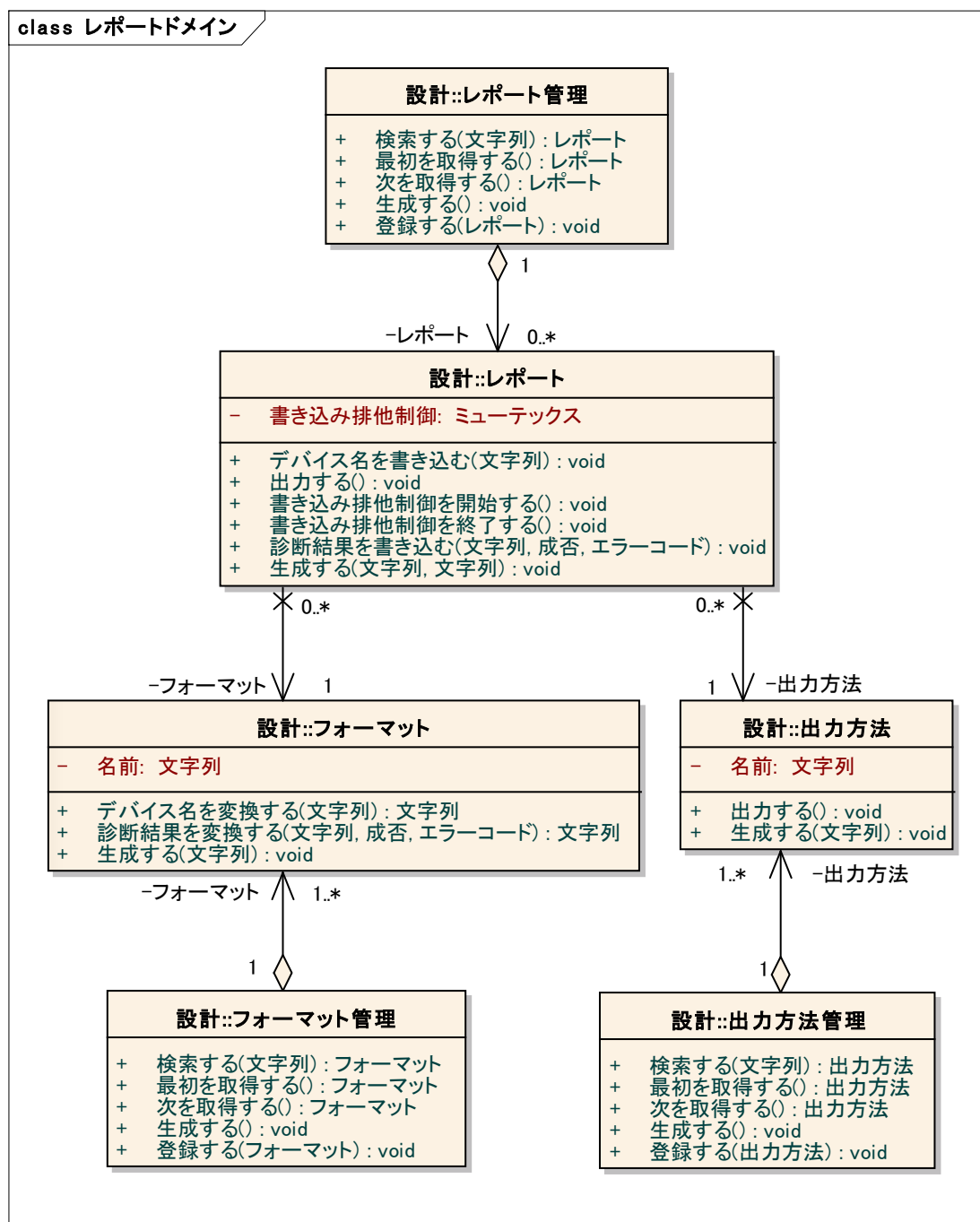


図17

レポート管理はレポートを集約し、管理します。

フォーマット管理はフォーマットを集約し、管理します。

出力方法管理は出力方法を集約し、管理します。

「ユーティリティ」のクラス構成

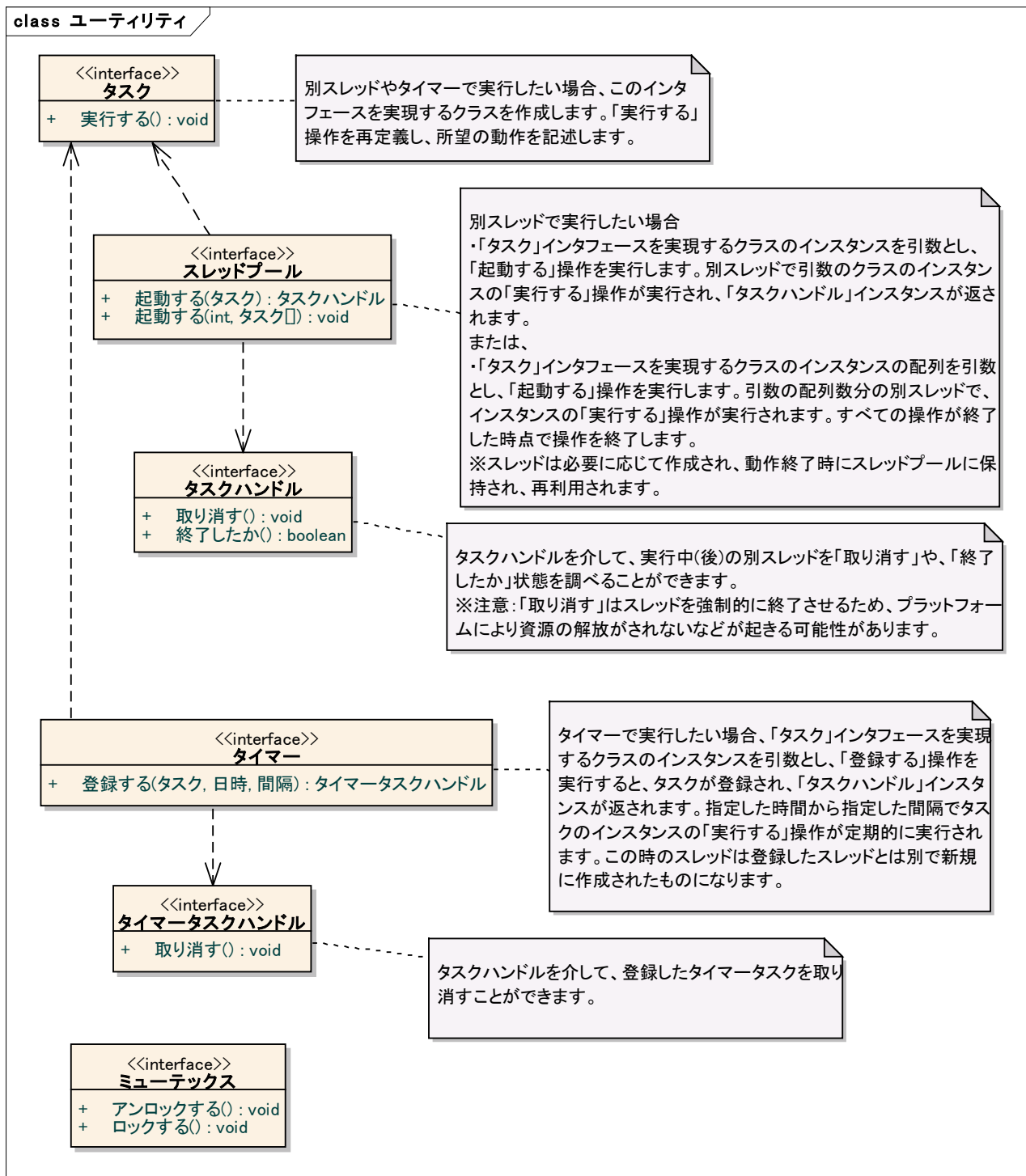


図18

動的モデル

PIM 設計モデルのクラス図を踏まえ、再度シーケンス図を作成します。

ユースケース「UC01:自己診断を行う」の相互作用

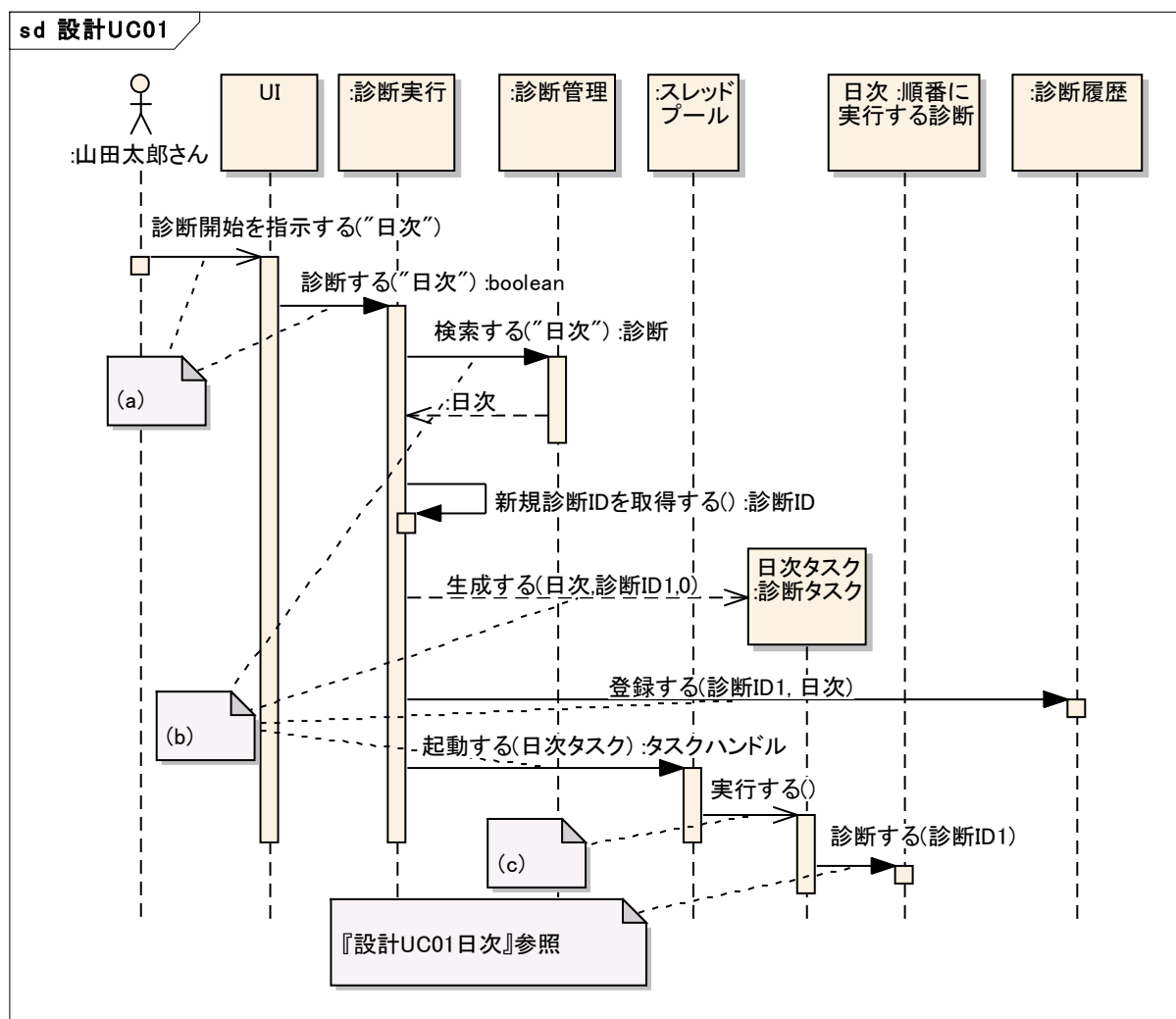


図19

- (a)アクターから診断開始の指示を受けると、UI は診断実行に「診断する」メッセージを送信します。
- (b)診断実行は診断管理から該当する診断インスタンス(「日次: 順番に実行する診断」)を検索し、診断タスクを生成し、診断履歴に登録し、生成した診断タスクをスレッドプールへ渡して診断を起動します。
- (c)スレッドプールは診断タスクへ「実行する」メッセージを送信し、「日次: 順番に実行する診断」に診断を実行させます。



- (a)「日次:順番に実行する診断」は「診断する」メッセージを受信すると、キャンセル(診断中止があるか)をチェックしながら、集約する診断インスタンスへ順次「診断する」メッセージを送信します。
- (b)(すべての診断を実行し終わると)レポートに排他制御を行い、出力させます。

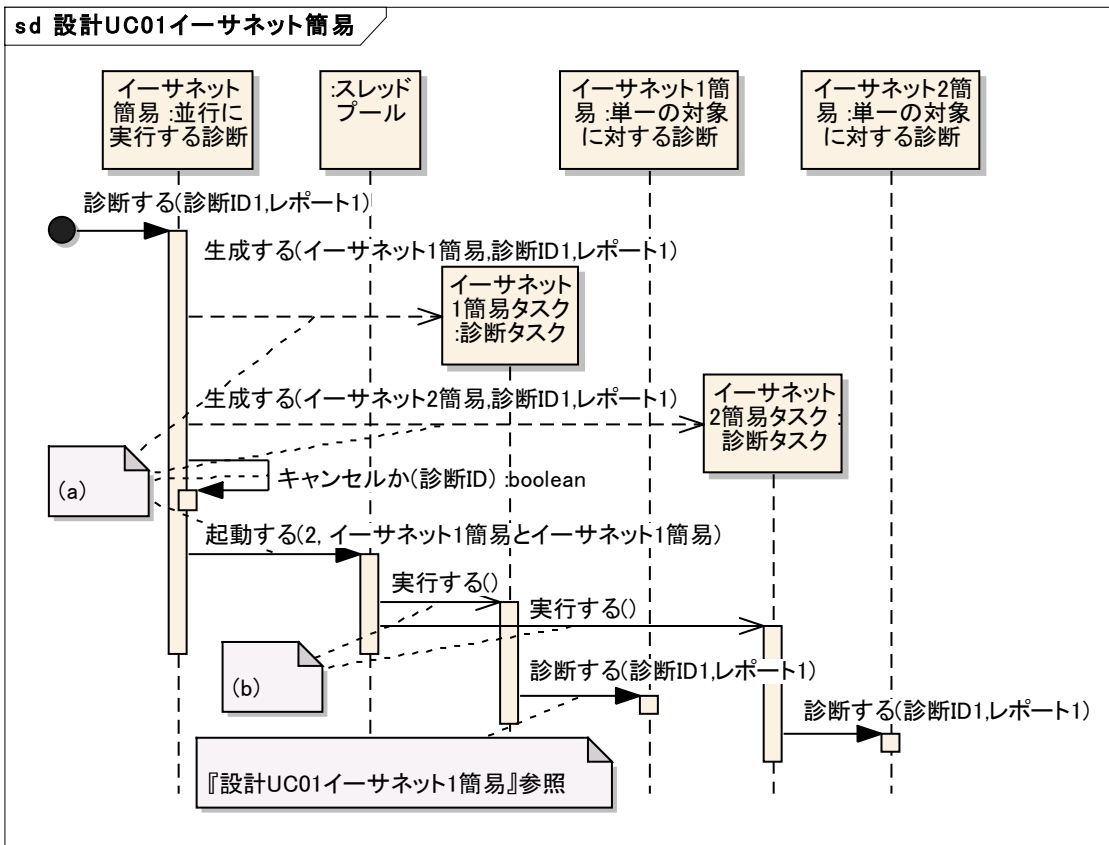


図21

- (a)「イーサネット簡易:並列に実行する診断」は「診断する」メッセージを受信すると、診断タスクを生成し、キャンセル(診断中止があるか)をチェックし、生成した診断タスクをスレッドプールへ渡して診断を起動します。
- (b)スレッドプールはスレッドを使用して、各診断を並列に実行させます。

sd 設計UC01イーサネット1簡易

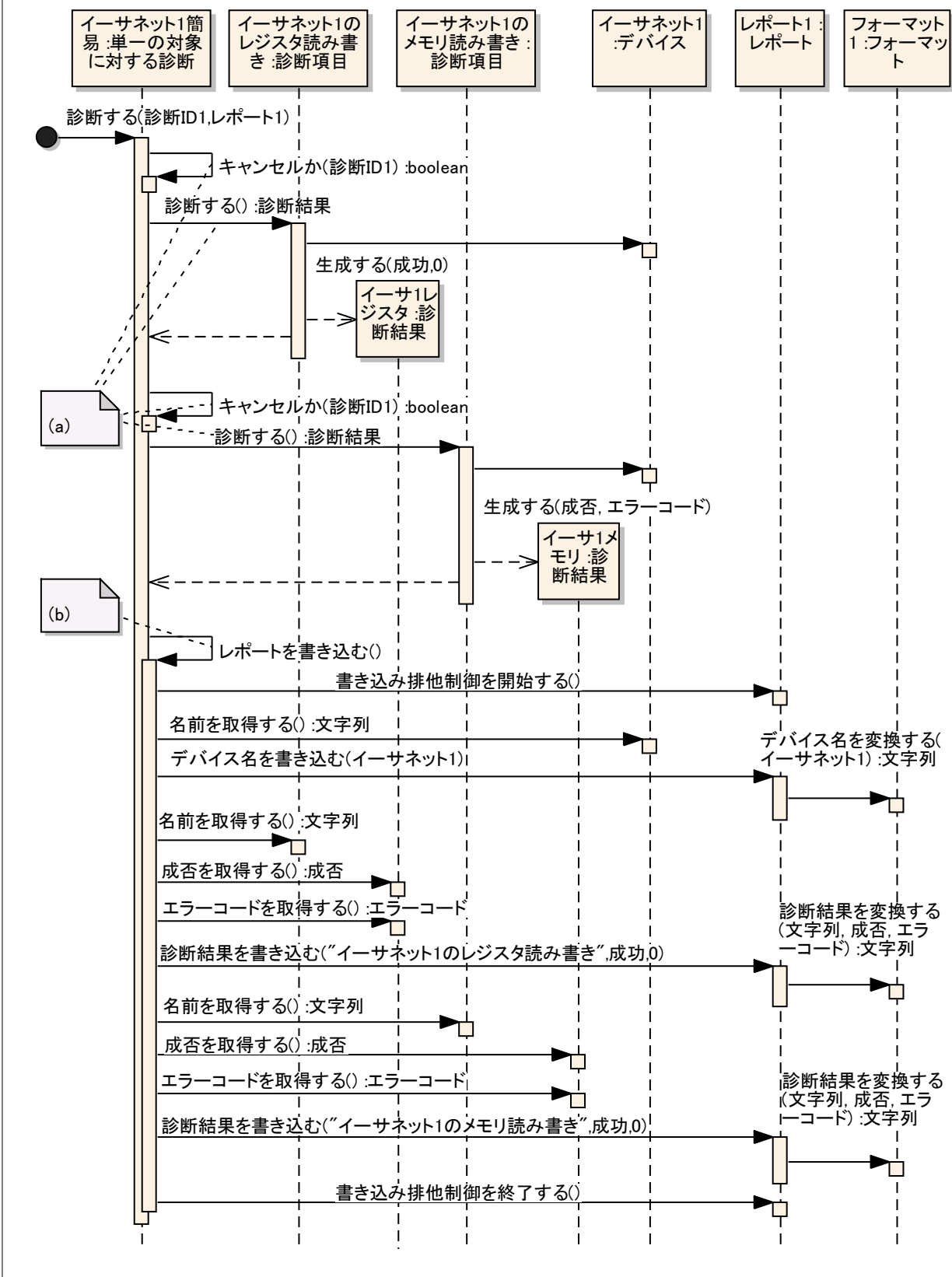


図22

(a)「イーサネット1簡易:単一の対象に対する診断」は「診断する」メッセージを受信すると、キャンセル(診断中止があるか)をチェックしなら、診断項目に診断を行わせます。

(b)(すべての診断を実行し終わると)レポートに書き込みます。

ユースケース「UC02:自己診断を中止する」の相互作用

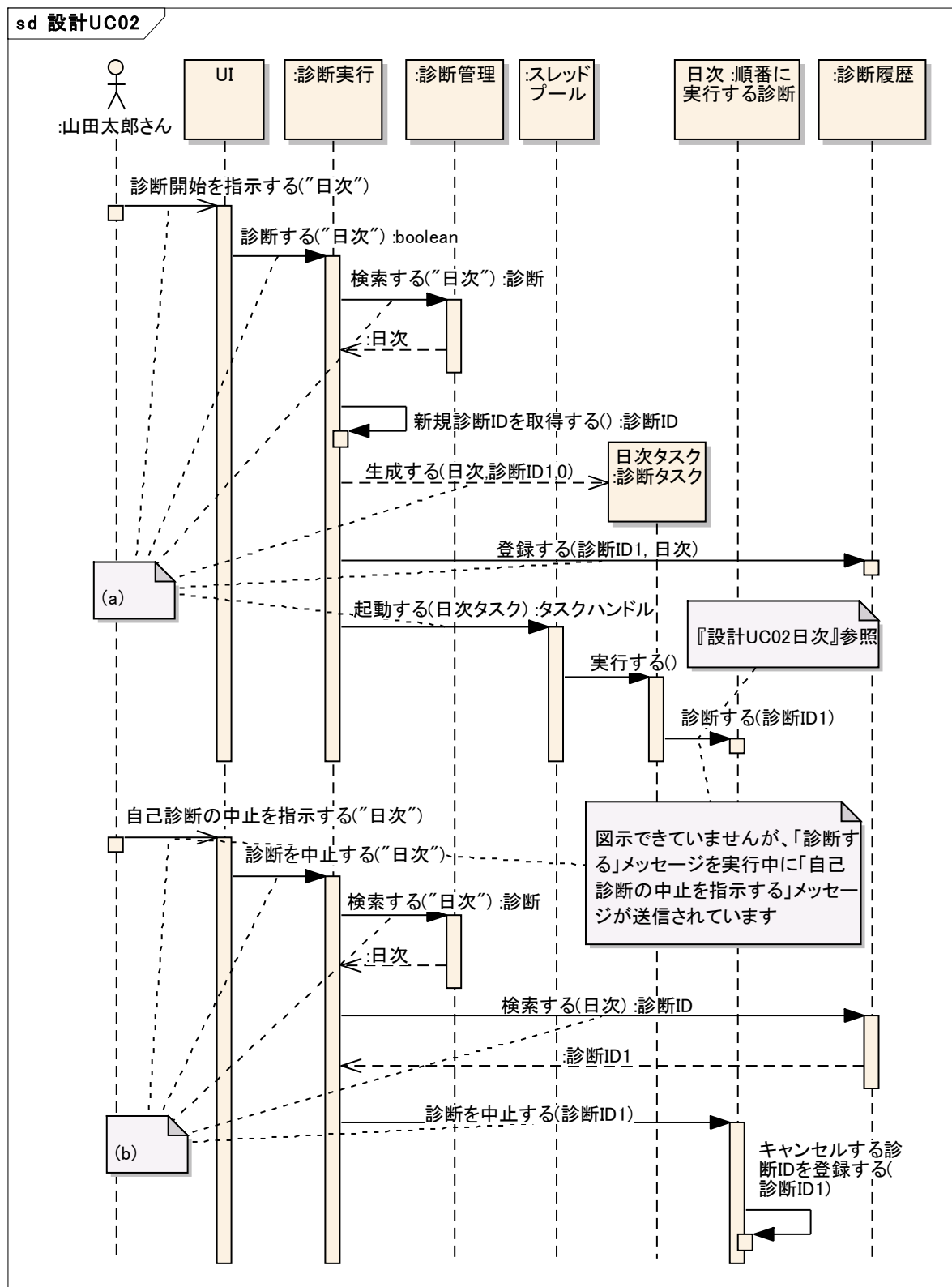


図23

- (a)アクターから診断開始の指示を受けると、ルーターは「ユースケース「UC01:自己診断を行う」のシーケンス図」と同様に動きます。
- (b)アクターから診断の中止の指示を受けると、診断実行は診断管理から該当する診断インスタンスを検索し、「診断を中止する」メッセージを送信します。

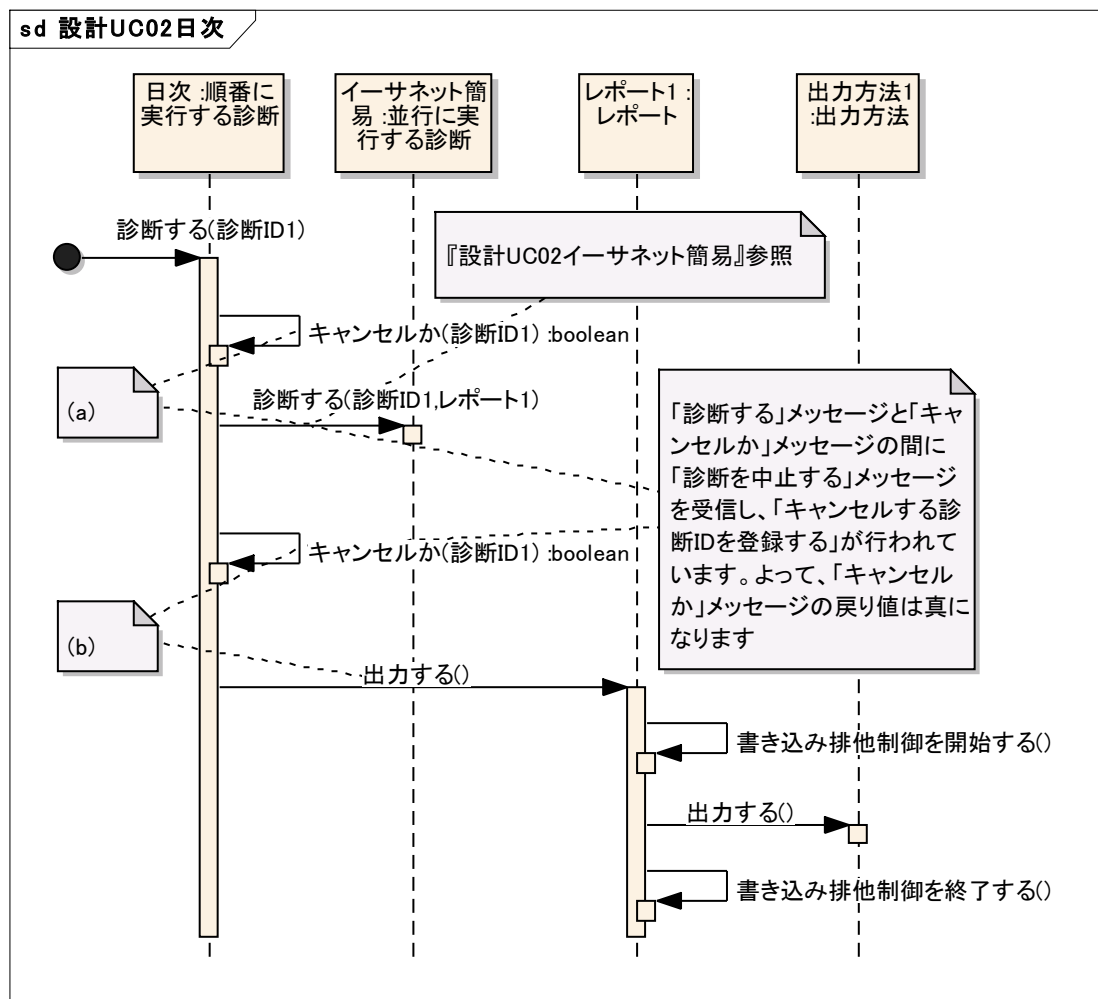
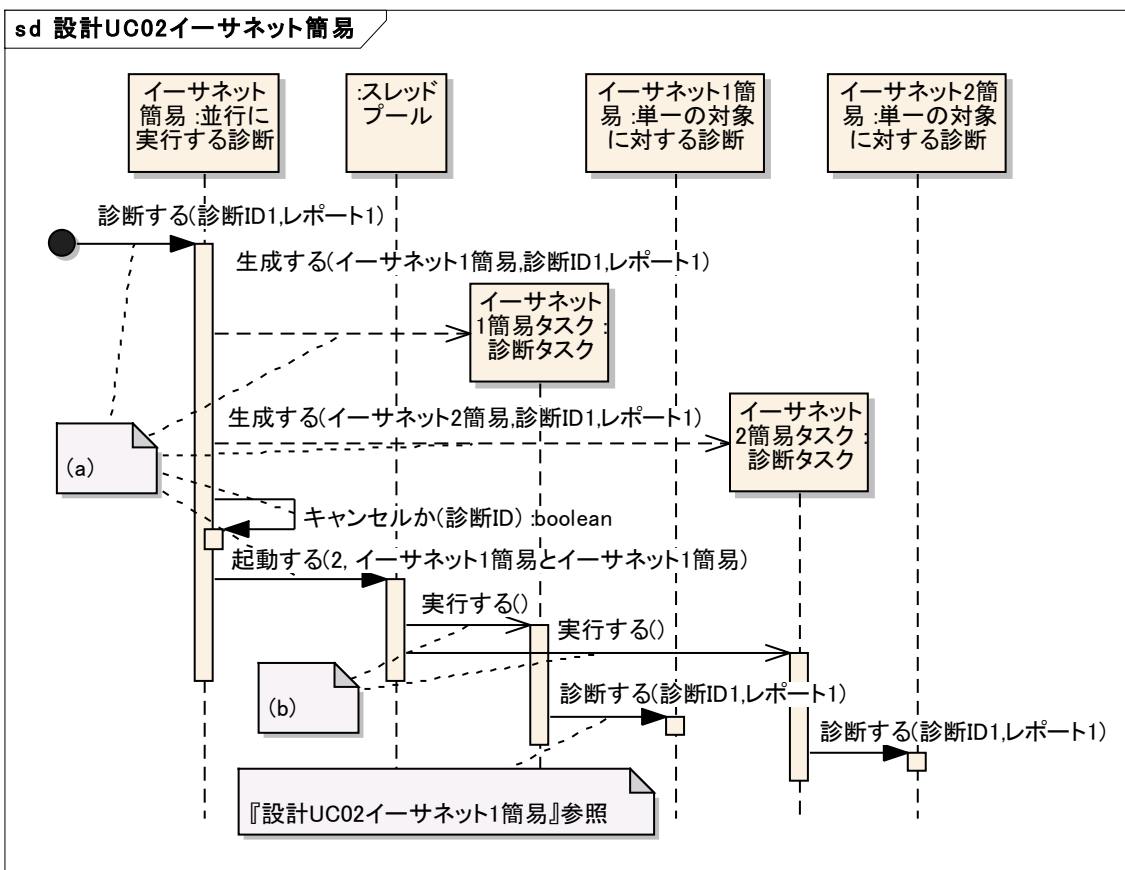


図24

- (a) 「日次:順番に実行する診断」は「診断する」メッセージを受信すると、キャンセル(診断中止があるか)をチェックしながら、集約する診断インスタンスへ順次「診断する」メッセージを送信します。
- (b) キャンセルがあると、「診断する」メッセージの送信を中止します。レポートに排他制御を行い、出力させます。



- (a) 「イーサネット簡易:並列に実行する診断」は「診断する」メッセージを受信すると、診断タスクを生成し、キャンセル(診断中止があるか)をチェックし、生成した診断タスクをスレッドプールへ渡して診断を起動します。
- (b) スレッドプールはスレッドを使用して、各診断を並列に実行させます。

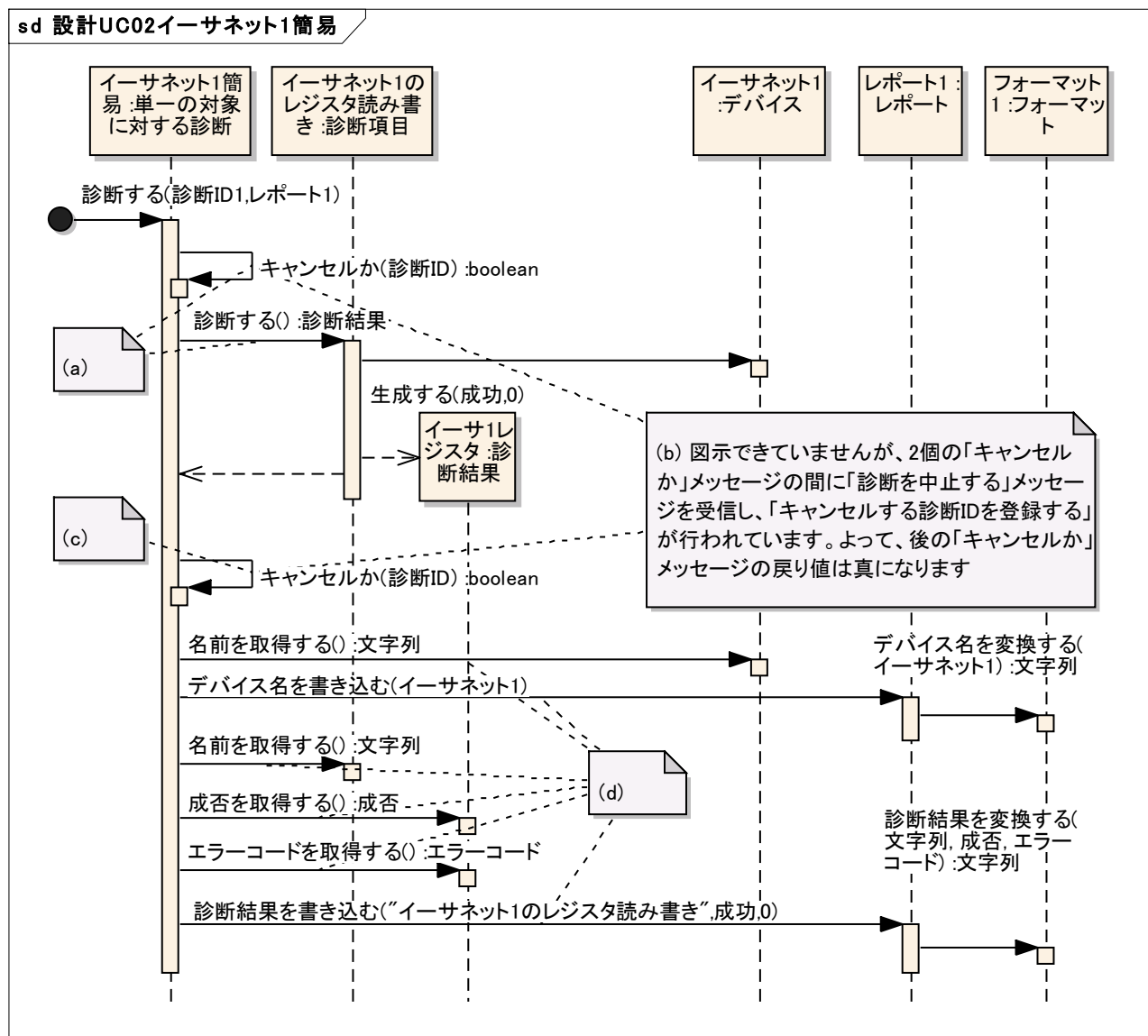


図26

- (a) 「イーサネット 1 簡易:単一の対象に対する診断」は「診断する」メッセージを受信すると、キャンセル(診断中止があるか)をチェックしなら、診断項目に診断を行わせます。
- (b) 図示できていませんが、「イーサネット 1 簡易:単一の対象に対する診断」からの 2 個の「キャンセルか」メッセージの途中で、「診断を中止する」メッセージを受信しています。
- (c) 2 個目の「キャンセルか」メッセージの戻り値が真になり、診断を中止します。
- (d) レポートを書き込みます。

※「キャンセルか」の判断は診断 ID を用いて行います。自身の診断 ID と「診断」クラスの「キャンセルする診断 ID」属性が合致すると、キャンセルとなり、「キャンセルか」メッセージの戻り値が真になります。このモデルでは「キャンセルする診断 ID」属性が 1 個のため、一時点でのキャンセルは一つの診断ですが、「キャンセルする診断 ID」を複数持つことにより、一時点で複数の診断のキャンセルができるようになります。

シナリオ「SC0401: 自己診断を設定する(1つの診断対象に対する診断)」の相互作用

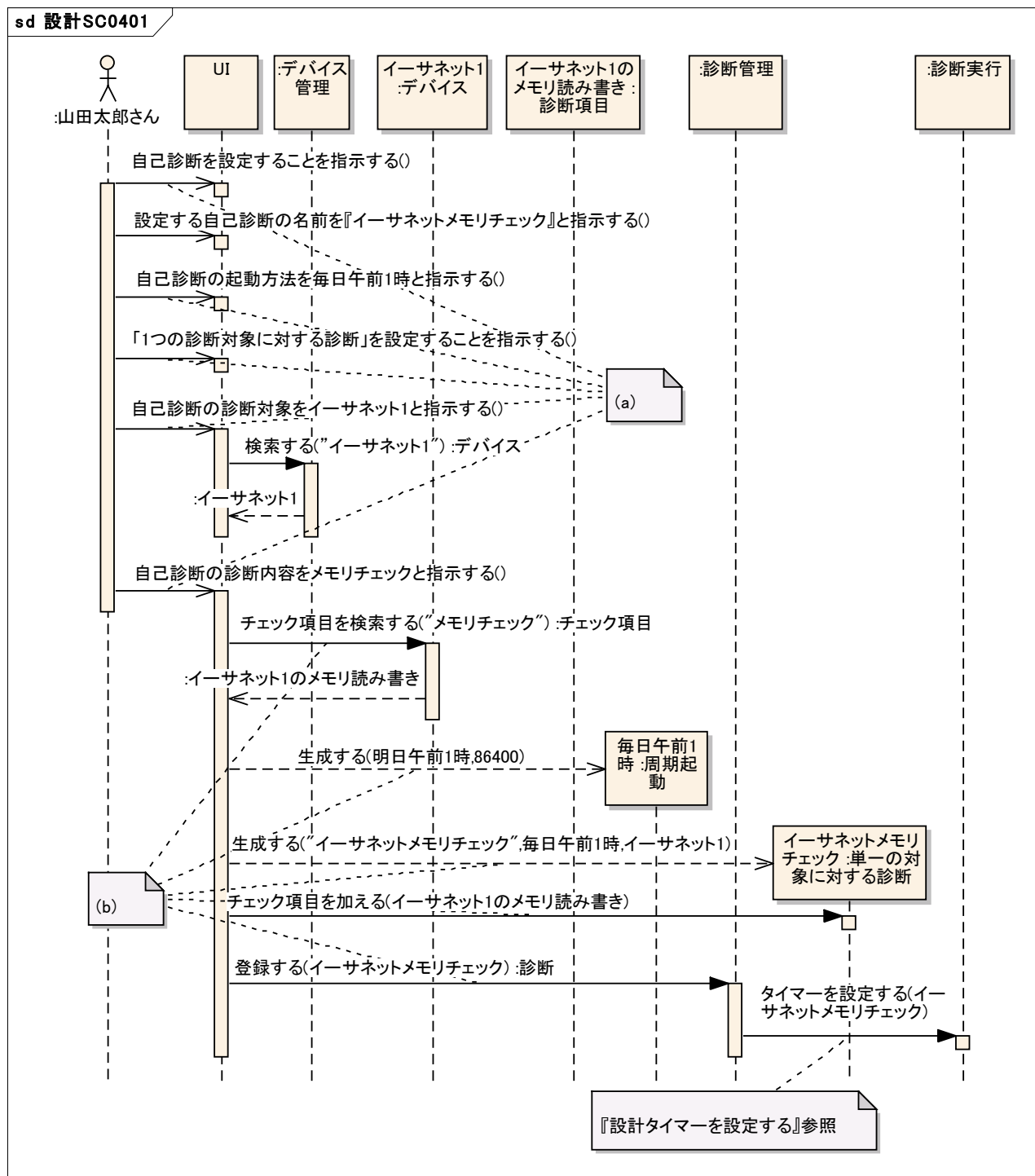


図27

(a)アクターはルーターへ自己診断の名前、起動条件、診断対象、診断内容を指示します。

(b)UI は起動条件と診断を生成し、関連付け、診断を登録します。

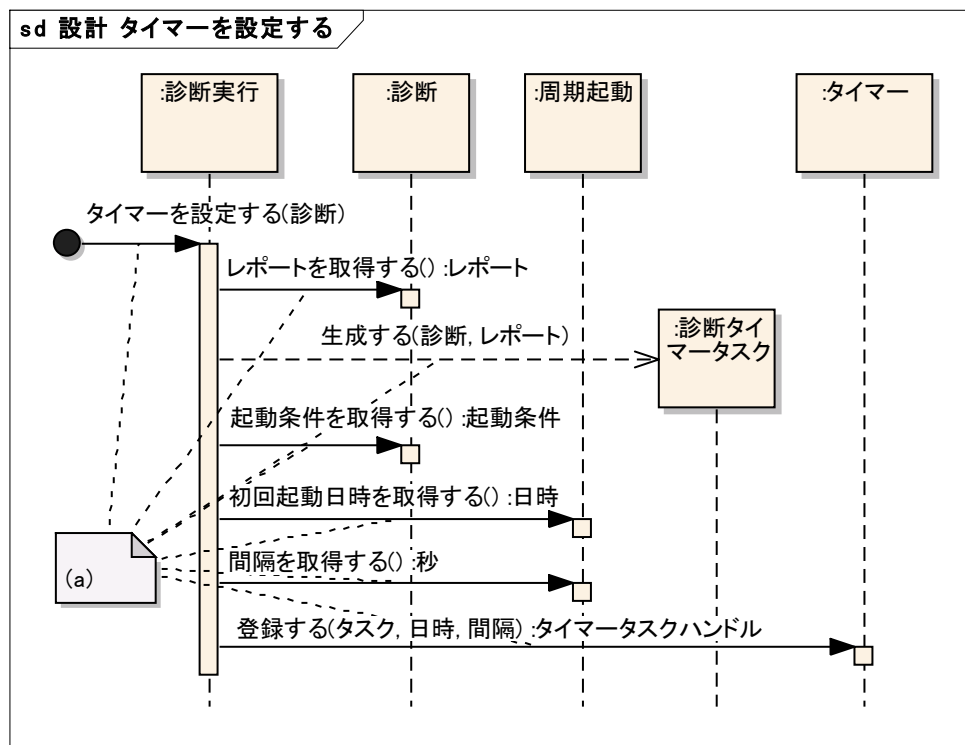


図28

(a)「診断実行」は「タイマーを設定する」メッセージを受信すると、必要な情報を取得し、診断タイマータスクを生成し、生成した診断タイマータスクをタイマーに登録します。

シナリオ「SC0403: 自己診断を設定する(既存の自己診断を組み合わせた診断)」の相互作用

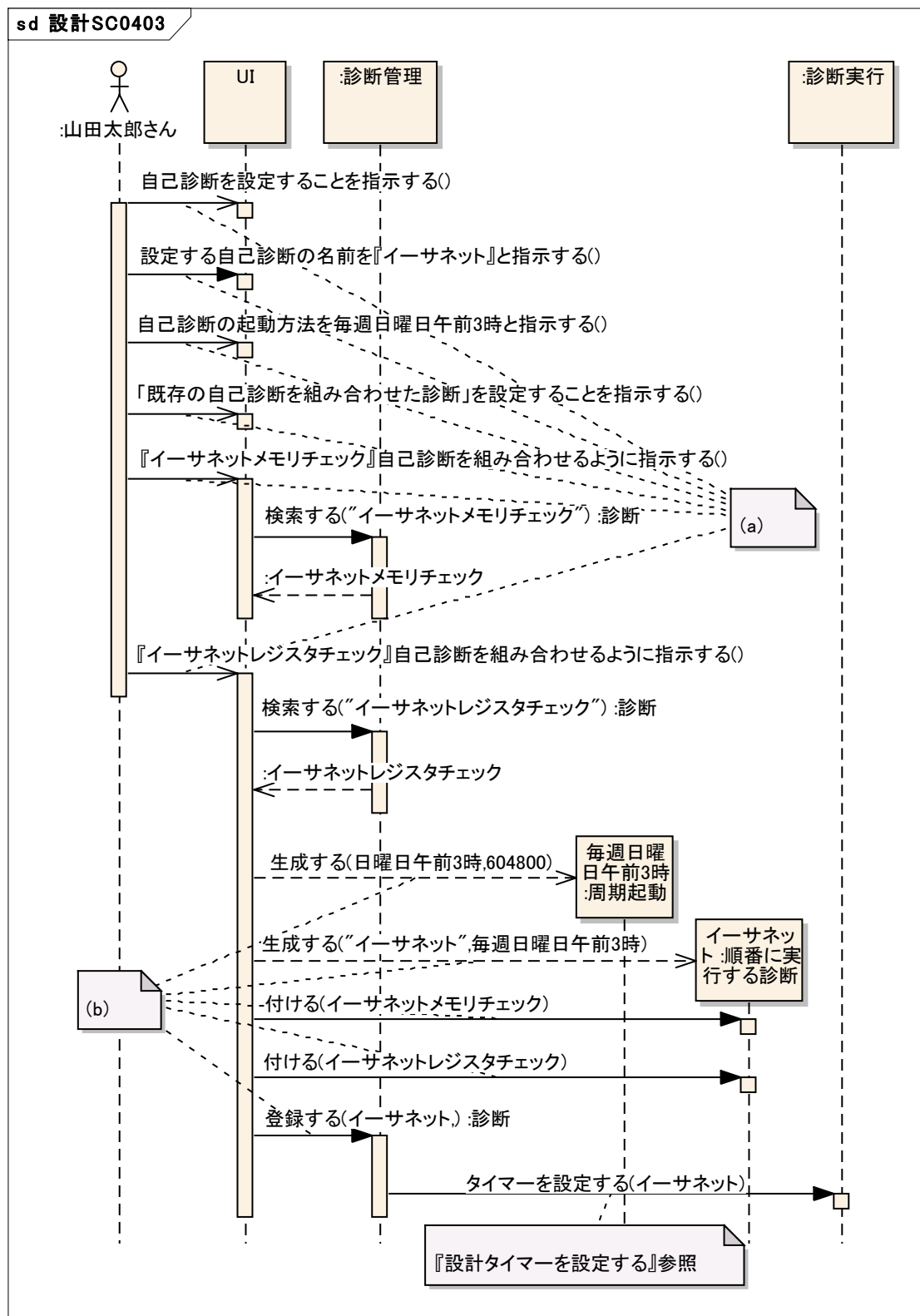


図29

(a)アクターはルーターへ自己診断の名前、起動方法、組み合わせる診断を指示します。

(b)UI は起動条件と診断を生成し、診断を複製し、関連付けます。

メタファを使ったモデル

モデリングのコンセプト

装置の自己診断機能をモデリングするにあたり、まずは診断の意味や目的を考え、日常生活の診断に適用できる概念モデルを作成してみます。診断についての理解を深め、概念を明確にすることで、分析モデルや設計モデルの作成に役立つと思います。

1. 診断って何だろう？

「診断」を国語辞書で調べてみました。(大辞林より)

しんだん 【診断】 (名) スル

- (1) 医者が患者を診察し、病状を判断すること。
 - ・ 「高血圧と診断する」
 - ・ 「診断を下す」
- (2) 物事を調べて欠陥がないかなどその状態を判断すること。
 - ・ 「企業診断」

つまり、診断とは病気や欠陥など悪いところがないか調べることです。装置の場合は故障箇所や不具合箇所がないか調べることです。悪いところに限らず、悪くなりかけているところや悪くなりそうなところも調べられると、よりよい診断といえるでしょう。

次に、「診断」の類義語を挙げてみます。これらも「診断」とみなしていいでしょう。

◆ 診断の類義語: 検診、点検、検査

ついでに、「自己」とは、国語辞書で調べるまでもなく「自分自身で」ということ。つまり、医者が自分を診断する、企業が自社を診断するということで、装置であれば装置自身で装置内部を診断することです。

2. 診断する目的は何だろう？

診断して悪いところが見つかったらどうしますか？

- ・医者に言われたなら、病気を治療する
- ・企業に欠陥があるなら、正す、改善する
- ・装置であれば、不具合箇所を修理、修復、交換する

悪いところをなおすのはなぜですか？

- ・人であれば、長生きしたいから。重症になると、介護が必要になったり、お金がかかったりするから。
- ・企業であれば、企業存続にかかわるから。
- ・装置であれば、業務に支障が出るから。

悪いところをなおすのは誰ですか？

- ・他者: 医者、業者
- ・自己: 自分、自社

以上のことから、問題箇所や予防箇所がないかを診断することで、自己修復あるいは他者に治療・改善・修理依頼するきっかけを知ることができ、診断対象を長生き・長持ちさせるのが目的と考えられます。

3. 身近な診断には何があるだろう？

身近にある診断、点検、検査を探してみました。

◆ 健康診断

- ・毎年健康診断を受ける。
- ・F社では、5年に1回人間ドックに変わる。40才以上は毎年人間ドック。
- ・健康診断と人間ドックでは診断項目が異なる。年齢により多少診断項目が異なる。

◆ 車検

- ・新車は3年目、以降は2年毎。車検の他に6ヶ月点検や12ヶ月点検もある。
- ・点検時期によって点検内容が異なる。

◆ 温水洗浄便座

- ・自宅の温水洗浄便座のコンセントプラグに「テスト」ボタンがある。
- ・ボタンを押すと正常に動作するか診断してくれるようだ。
- ・手洗い時に水が跳ねたり湿気を含んでショートしたりする心配があるからだろうか？

◆ 分電盤

- ・自宅の分電盤の蓋をあけると「テスト」ボタンがある。
- ・漏電やショートしていないか診断してくれるのだろう。

◆ 消防設備点検

- ・自宅マンションでは、年に2回、火災報知器が正しく動作するか調べに来てくれる。
- ・会社でも、年に2回くらい消防設備を点検しているのを見かける。

4. 診断に関する共通項目には何があるだろう？

前述の身近な診断(点検、検査を含む)において、共通する項目を挙げてみます。

対象物 共通項目	人	車	温水洗浄便座	分電盤	消防設備
診断名	健康診断 人間ドック	車検 定期点検	テスト	テスト	消防点検 防災点検
診断時期	年 1 回 望んだとき	初回 3 年 以降 2 年毎	ボタンを押した とき	ボタンを押 したとき	半年毎
診断対象	本人、子供、社員	車	電気設備	電気設備	消防設備
診断依頼者	本人、親、会社	所有者	家人	家人	本人、管理会 社、企業
診断場所	病院、健康管理室	車ディーラー、 車検場	設置場所	設置場所	設置場所
診断実施者	医者	メカニック	装置	装置	業者
診断項目	血液、血圧、視力、聴 力、胃、肺、心臓など	エンジン、ブレ ーキ、オイル、 タイヤなど	ショート	漏電、ショ ート	熱検知、煙検 知、防災扉、シ ャッター等
診断結果	健康ランク、問題部 位、状態など	問題なし、 要修理箇所、 予防修理箇所	ブザー？	ブザー？	異常なし 異常箇所

5. 診断シナリオ

共通項目を使って、診断するときの手順(シナリオ)を考えてみます。

- ◆ 登場人物(アクター)は「診断依頼者」「診断実施者」「診断対象」の 3 人。
- ◆ 本人が本人の診断を依頼すれば「診断依頼者」と「診断対象」は同一です。
- ◆ 「診断依頼者」は、「診断時期」になった「診断」がないか定期的に確認し、あれば「診断実施者」に「診断対象」の「診断」を依頼します。
- ◆ 「診断時期」に関係なく、「診断対象」が調子悪くなったり、なんとなく気になったときに「診断依頼者」が「診断実施者」に「診断対象」の「診断」を依頼することもあります。
- ◆ 「診断実施者」は、「診断項目」毎に「診断対象」を「診断」し、「診断結果」を記入して「診断依頼者」に通知します。
- ◆ 「診断対象」は「診断部位」を診せます。
- ◆ 「診断依頼者」は、「診断結果」を確認し、問題があれば治療や修理などを施します。



- 47

「診断」の相互作用

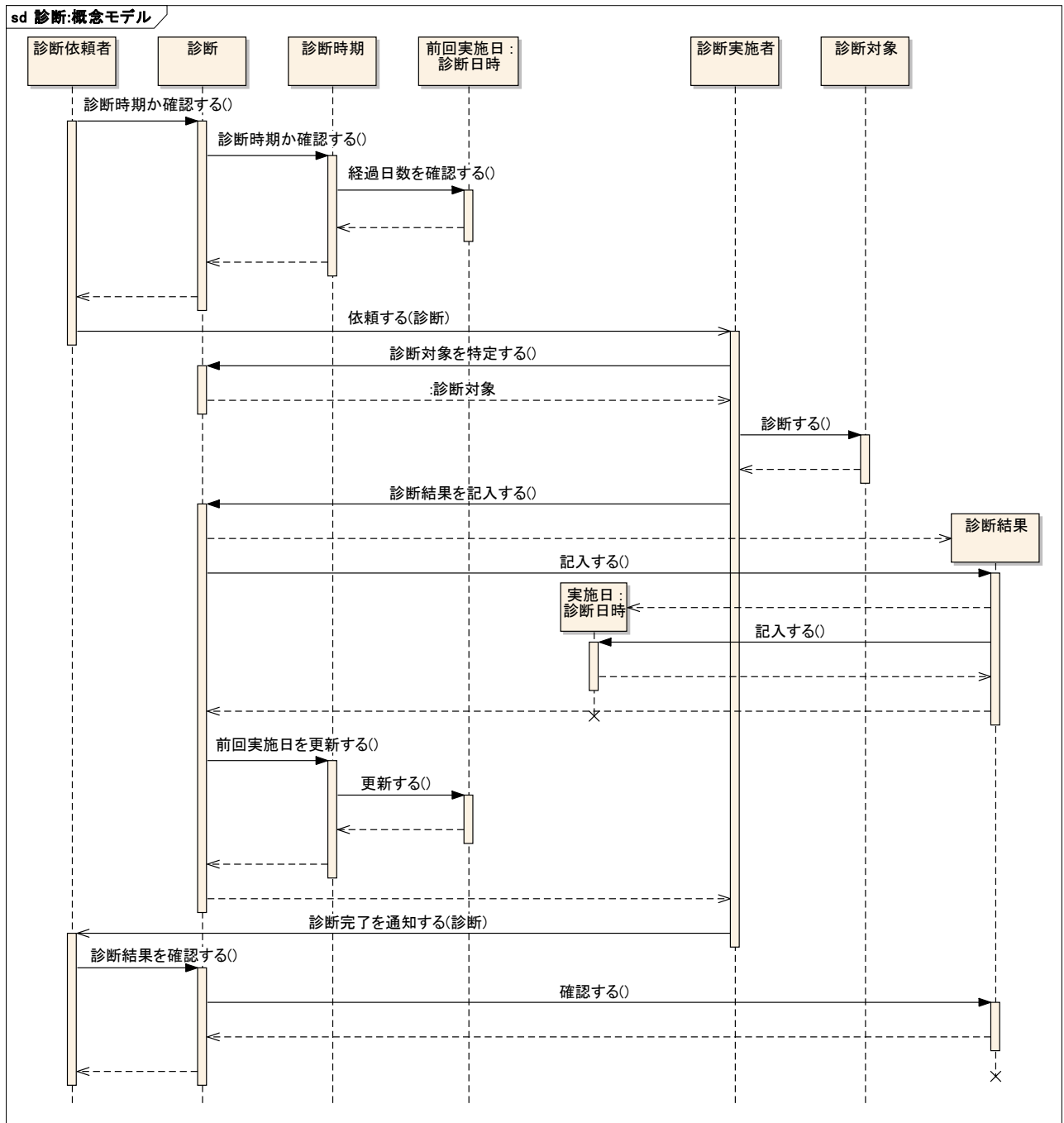


図31

処理の流れをシーケンス図で書いたものです。

シーケンス図により、クラスに追加・変更する操作がわかってきます。

分析モデル

要求仕様で出てくる要素として「自己診断」「診断対象」「診断項目」「診断結果」があります。それらの関係に重点を置き、概念モデルを参考にしつつ、ユースケースをできるだけ忠実にモデル化してみます。

なお、診断対象がどのように診断するのかについては関知せず、並行実施するか順次実施するかも診断対象にまかせることにし、診断を実施することによって診断結果がレポートされることに重点を置きます。

静的モデル

概念モデルを元に分類したパッケージ

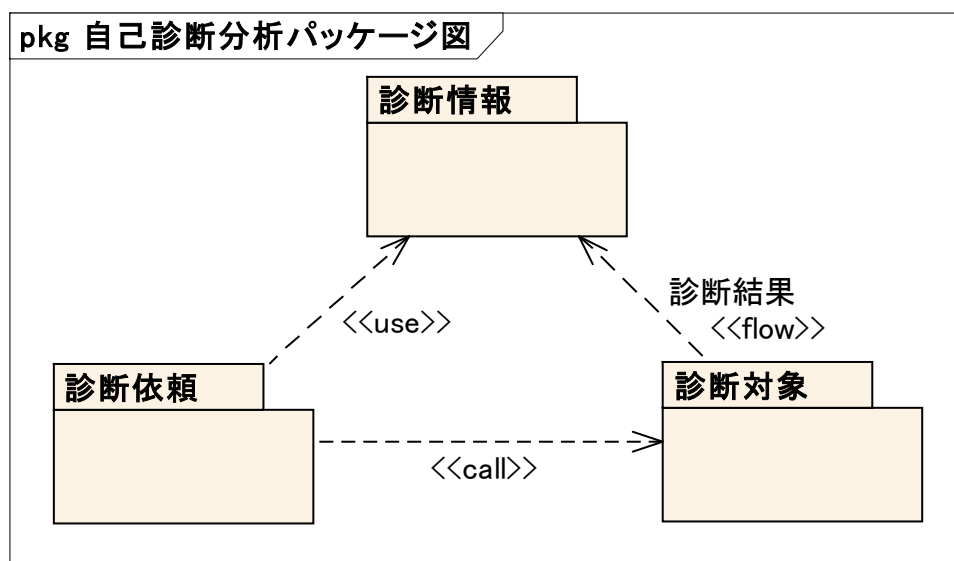


図32

概念モデルで出てきた「診断依頼者」は「診断依頼」とします。ここには、管理者からの「自己診断」の要求を受け付けて、「診断対象」に依頼するまでの部分を担当します。

自己診断では、「診断対象」自身が「診断実施」するので、「診断実施」は「診断対象」にまとめます。

「診断依頼」と「診断対象」の間でやりとりされる情報を「診断情報」としてまとめます。ここには「診断」と「診断結果」が入ります。

概念モデルでの「診断時期」は、自動診断の設定情報として「自己診断」に含めることにします。「診断日時」は要求仕様には出てこなかったので扱わないことにします。

それらを踏まえてクラス化していきます。

「自己診断」には、機器管理者の要求により起動される「手動診断」と、予め設定された「診断条件」に合わせて起動される「自動診断」があります。いずれも「自己診断」のサブクラスです。

「自己診断」を実施する際には、「診断結果」をレポートするための「診断出力」を用意しておきます。コンソール出力、ロギング出力、LED 点滅など、いろいろな「診断出力」の実装が考えられます。

「診断対象」には「機能ユニット」「デバイス」「アプリケーション」があります。いずれも「診断対象」のサブクラスです。「機能ユニット」はいくつかの「デバイス」から構成されます。

「診断対象」には「診断項目」が決まっています。「自己診断」を実施する毎に「診断結果」がレポート(通知)されます。「診断項目」に応じた診断が行われるので「診断項目」も「診断対象」のサブクラスにして診断の実施と中止ができるようにしておきます。

「診断対象」で診断を実施するときは、何の診断なのかを表す診断名と、「診断結果」をレポートできる「診断」を渡すことにします。「診断」は並列動作可能なので、「診断結果」は非同期で通知できるようにしておきます。「診断結果」が通知されたら診断完了したとみなすことができます。つまり、「診断」には診断実施中なのか診断完了したのかといった状態が存在することになります。

要求仕様にできるだけ忠実にしたためこのようなクラス図になりましたが、「診断」自身が診断実施できた方がいいなどの見直しは PIM 設計で行うことにします。

ユースケース記述に記載されたインスタンスに、関連しそうなインスタンスを加えたインスタンス図を以下に示します。

診断毎に診断結果が生成され、診断出力されることになります。

オブジェクト構成

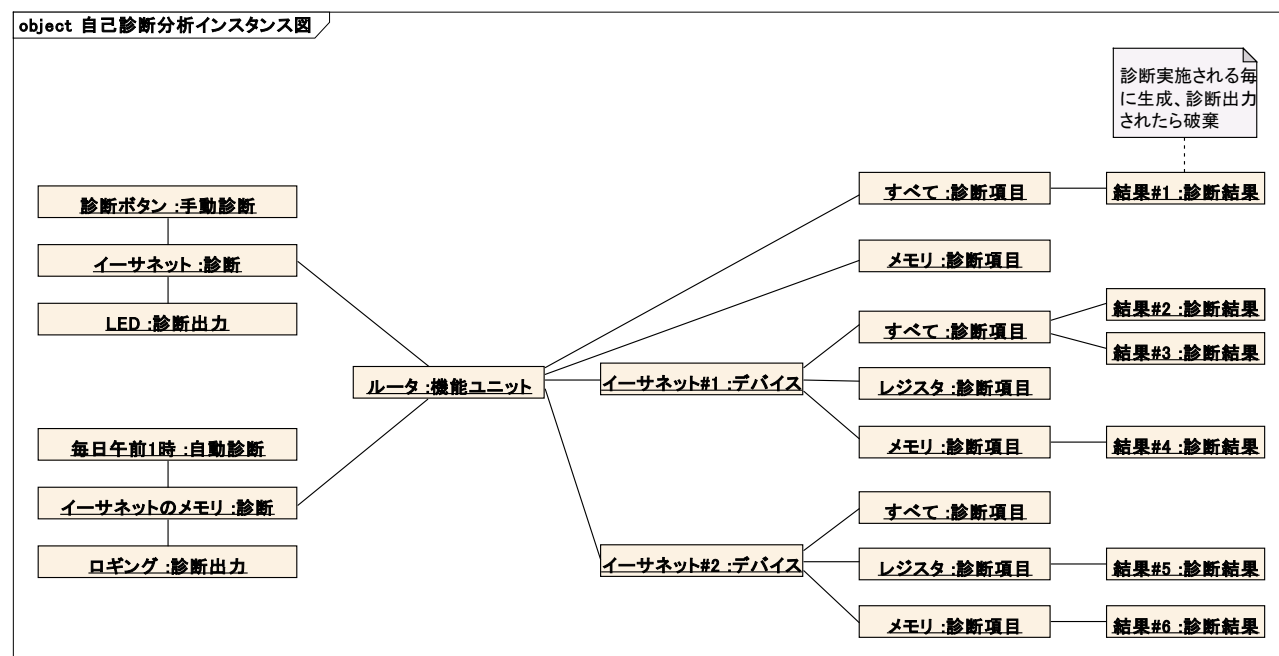


図34

動的モデル

診断を実施および診断を中断するときの相互作用

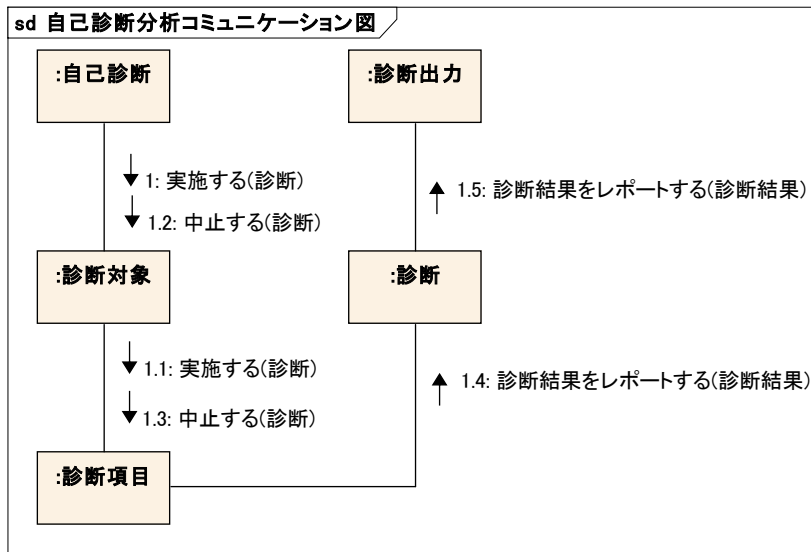


図35

管理者の指示により自己診断を実施し、診断を中止するときの相互作用

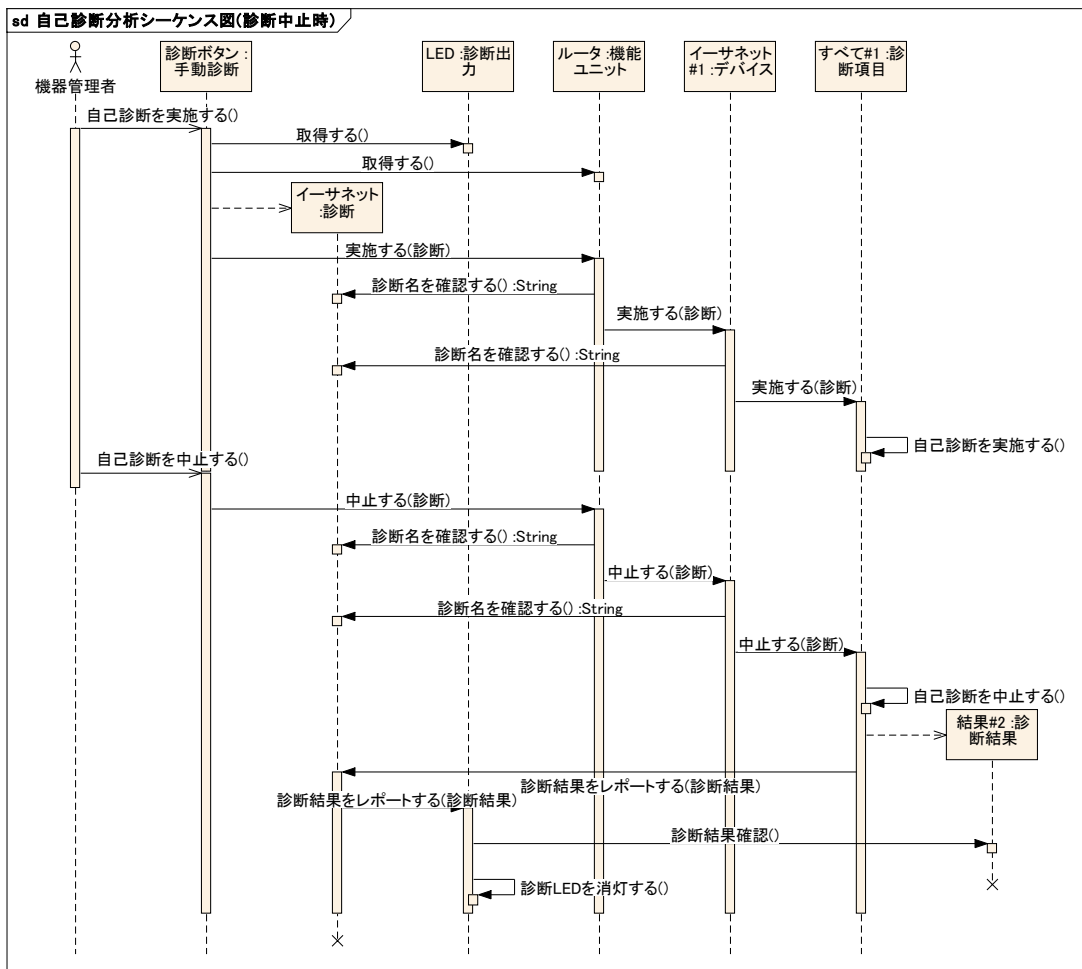


図36

PIM 設計モデル

分析モデルの関係性を崩さないように、依存性をできるだけ排除したコンパクトなモデルにします。

静的モデル

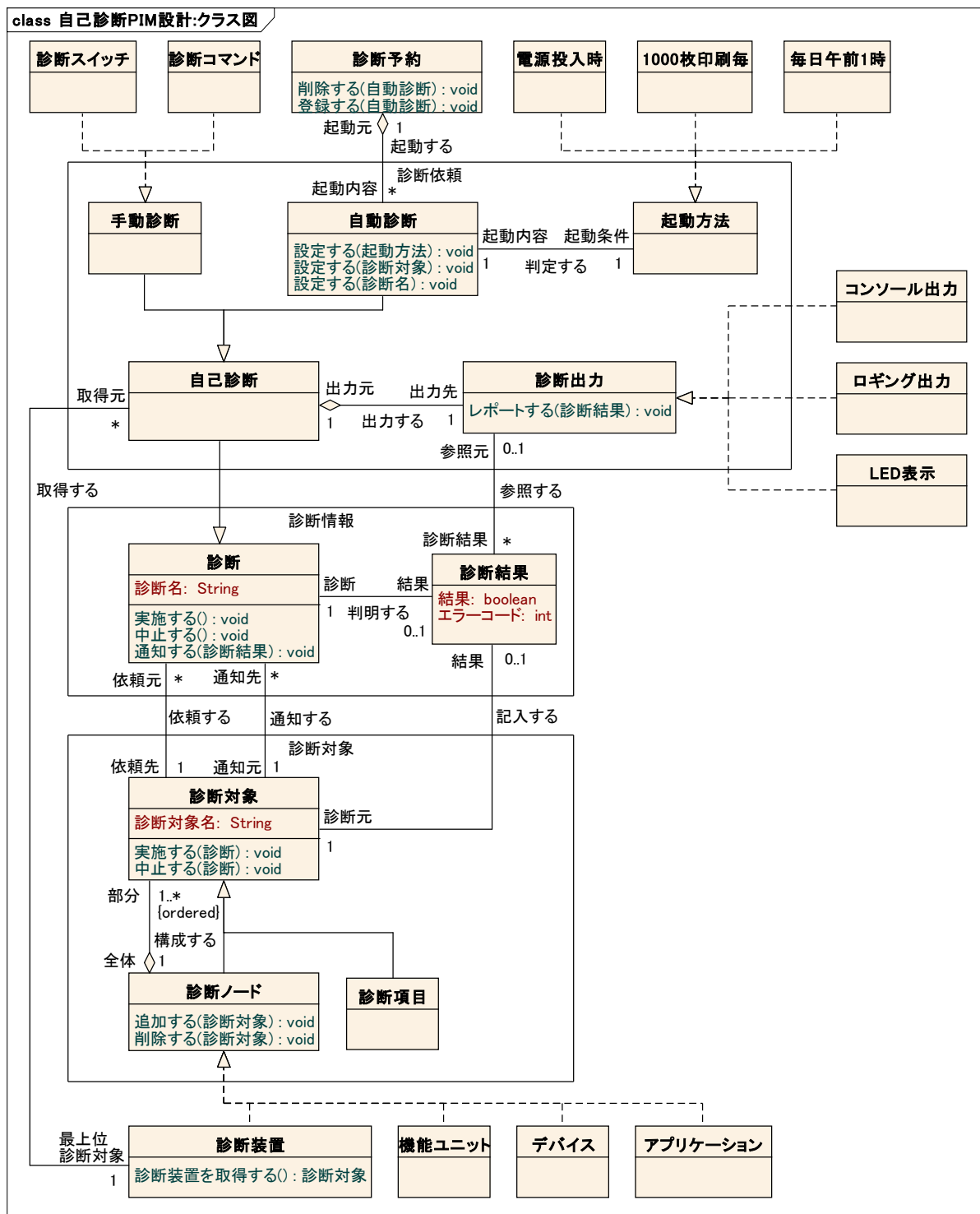


図37

「自動診断」は、「診断予約」として複数登録できるようにしました。ただし、「診断予約」は大抵の装置に備わっていると思われるスケジュール機能を利用することを想定し、本モデルでは詳しく取り上げないことにします。

分析モデルでは「自己診断」から「診断対象」に診断実施を指示していましたが、パッケージの独立性を高めるため、「診断」で診断実施できるようにし、「診断依頼」パッケージと「診断対象」パッケージを完全分離させました。

「診断」は、生成したあと 1 度だけ診断実施して破棄することができますが、「診断」をひとつ生成したあと何度も診断を実施して最新の診断結果を受け取ることもできます。

分析モデルでの「診断対象」は「診断項目」を保持していましたが、「機能項目」自体も「診断対象」のサブクラスであり、その構造がファイルシステムの「ディレクトリ」と「ファイル」のツリー構造に類似しているので、「ディレクトリ」相当を「診断ノード」とし、「ファイル」相当を「診断項目」としました。これにより、診断対象がファイルのディレクトリ名のように指定でき、以下のように診断対象を文字列で指定できるようにします。

- すべての診断: "*"
- メイン基板のすべての診断: "main/*"
- メイン基板のメモリ診断: "main/memory"
- すべての基板のメモリ診断: "*/memory"

なお、「診断ノード」から下位の「診断対象」に診断を依頼するには、「診断」オブジェクトが必要になります。依頼元から渡ってきた「診断」をそのまま渡すと、複数の「診断結果」が「診断出力」に渡ることになり、診断経過が出力されるような動作になります。ひとつの診断毎に「診断結果」を待ち合わせるような場合には、一時的な「診断」オブジェクトを生成して診断依頼し、診断結果を待ち合わせるようにする必要があります。

オブジェクト図は分析モデルと同等です。

動的モデル

メモリ診断を実施するときの相互作用

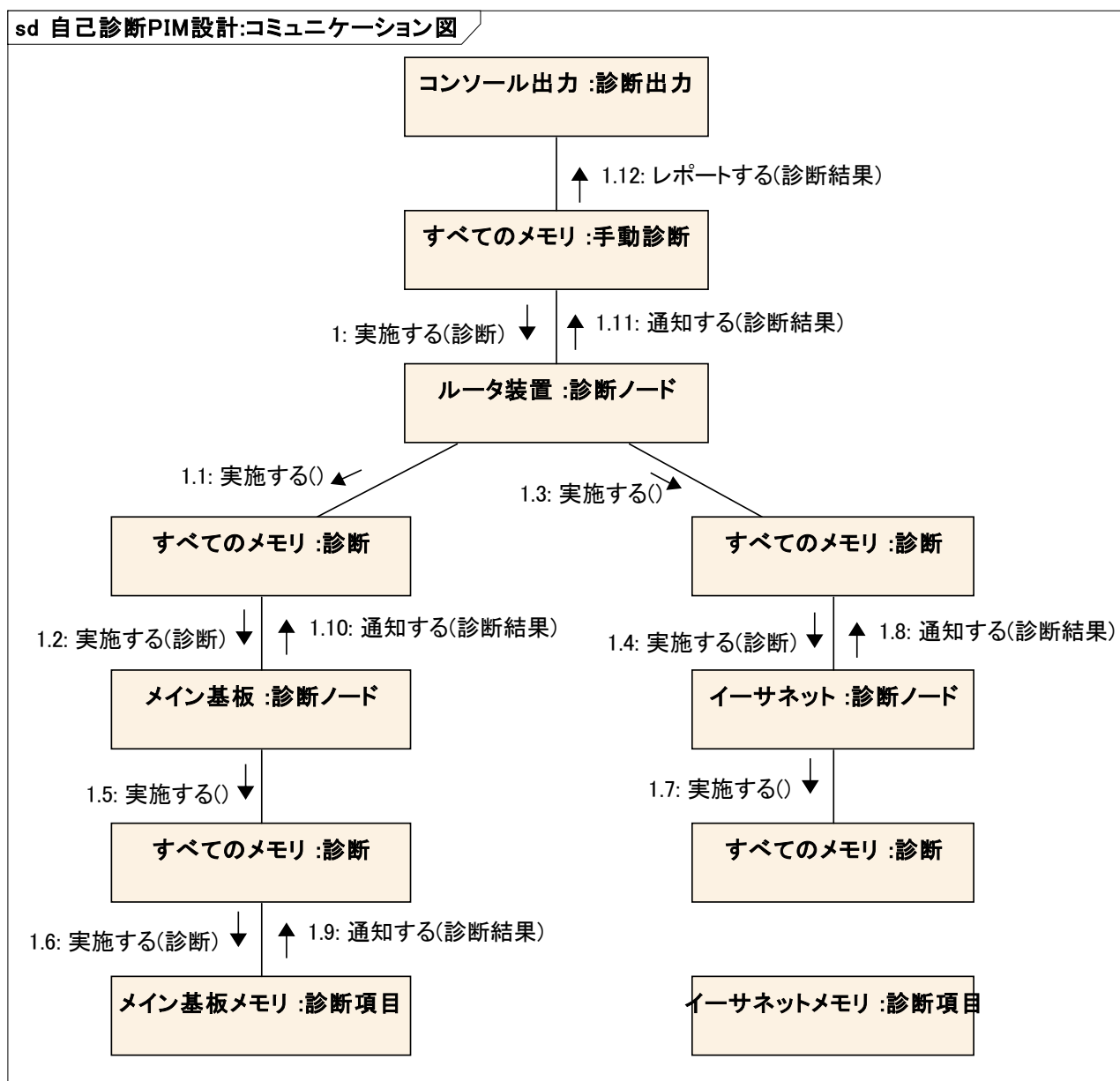
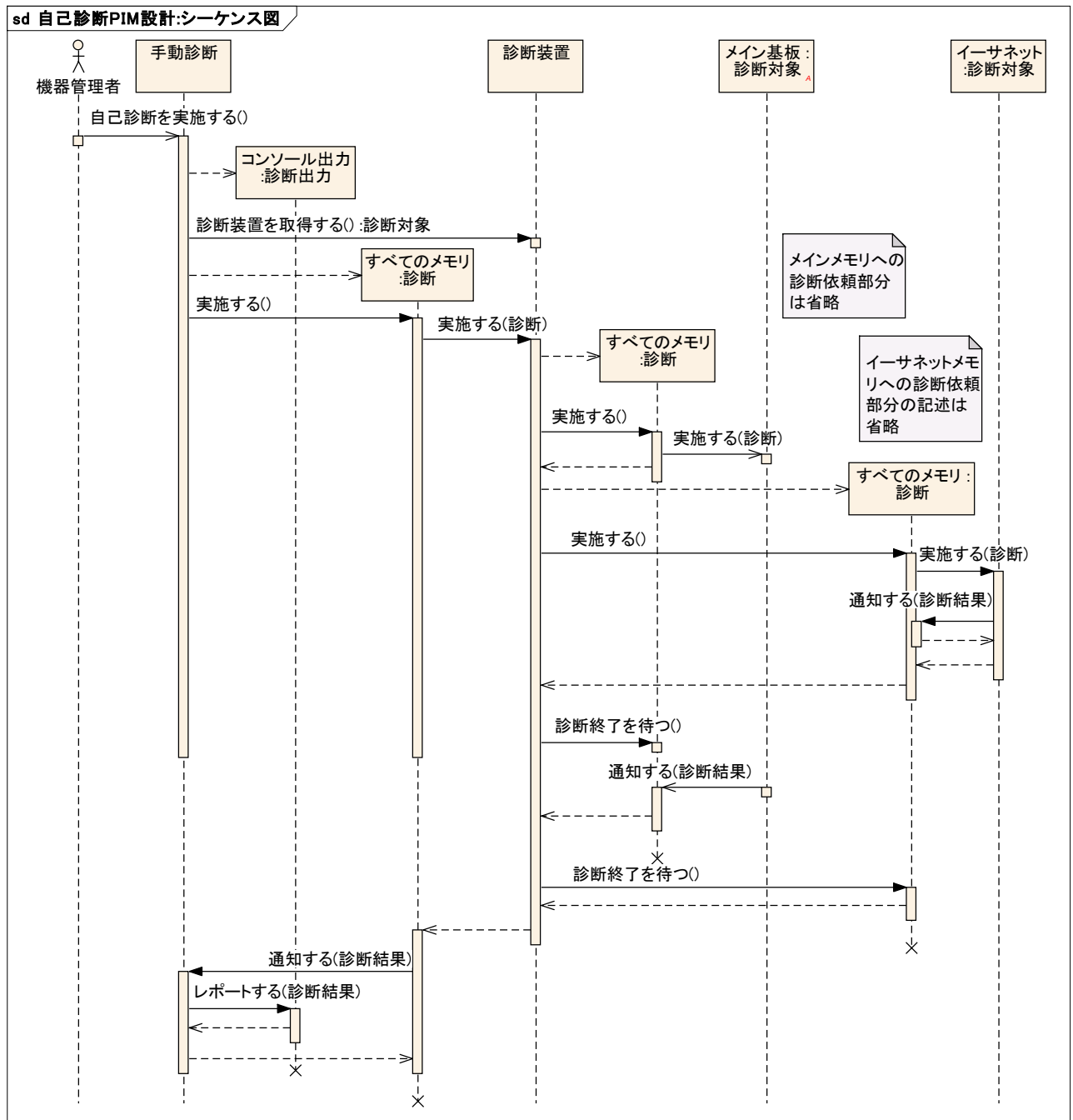


図38

メイン基板のメモリ診断とイーサネットのメモリ診断は並列動作し、イーサネットのメモリ診断の方が先に終わる順序にしてあります。

メイン基板とイーサネット基板それぞれに搭載されたメモリを診断するときの相互作用



メイン基板での診断はハードウェアで行われ、診断依頼は即時復帰し、診断完了は割り込みで通知される非同期型を想定しました。ソフトウェアで行う場合でも、スレッド生成して非同期型にすることもできます。

サブ基板での診断はソフトウェアで行われ、診断実施は診断終了まで復帰しない同期型を想定しました。診断装置もソフトウェア実装なので同期型にしました。

診断対象が診断結果を生成する記述は省略しました。診断結果をオブジェクト(クラス)にせずエラーコードを直接診断に通知するように仕様変更してもいいと思います。

診断対象クラスの状態遷移

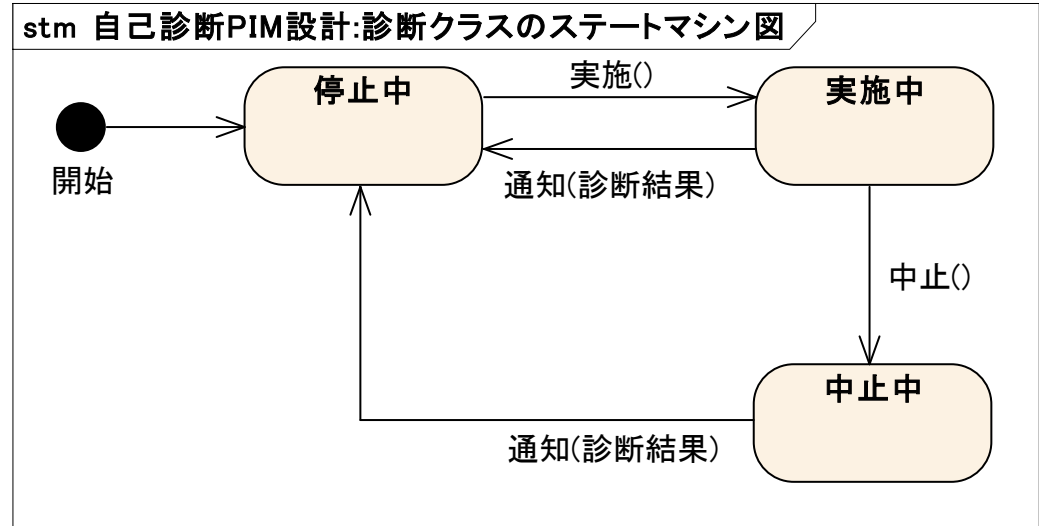


図40

診断対象クラスの状態遷移

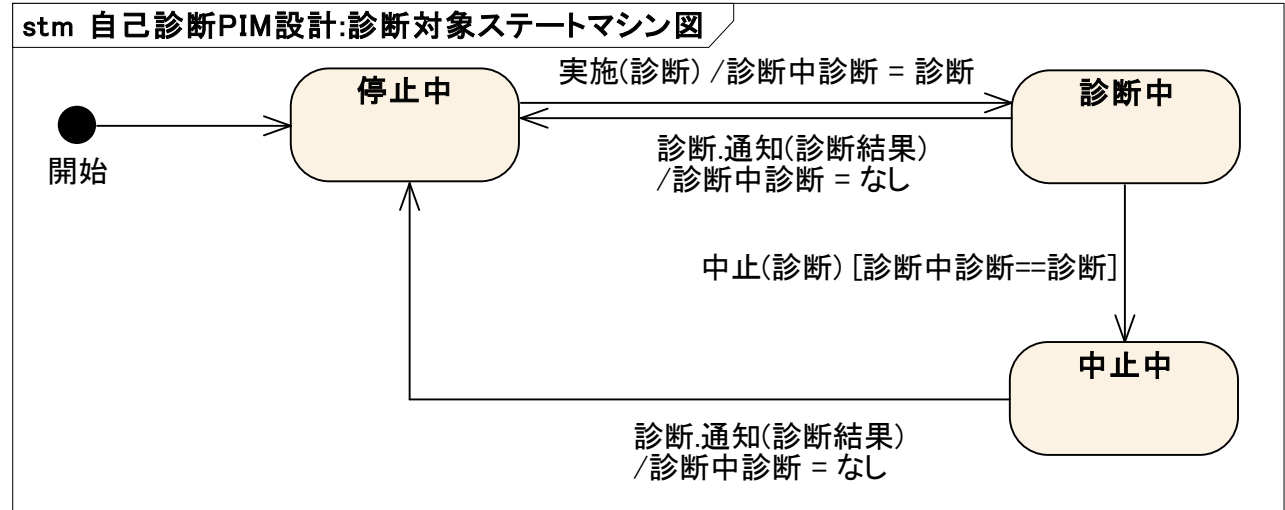


図41

組込み分野のための UML モデル解説書
「自己診断」

発行日 2010年(平成22年)10月1日
第二版発行 2012年(平成24年)5月17日
発行者 UMLTP, Japan
編 著 組込みモデリング部会
印刷

UMLTP, Japan
東京都渋谷区代々木1丁目22番1号
<http://www.umltp-japan.org/>