

動的プログラム呼び出しライブラリ

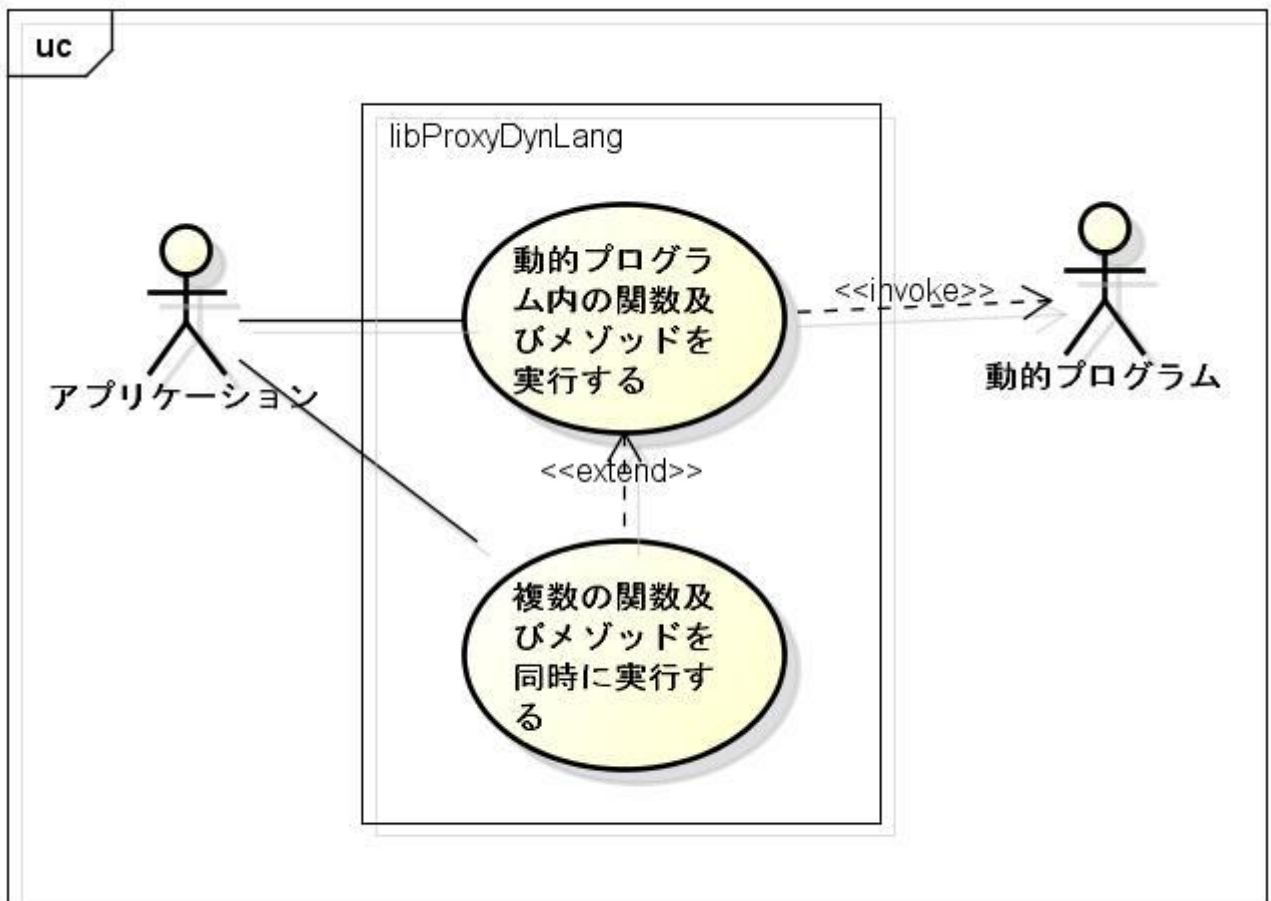
## 1. ライブラリ要件の定義

### (ア) ライブラリ概要

#### ① 背景と目的

アプリケーションから、動的プログラムのライブラリを使って、関数及びメソッドを実行するライブラリです。

#### (イ) システム化の範囲



## 2. ライブラリ機能要件の定義

### (ア) システム機能要件

ユースケース名		動的プログラム内の関数及びメソッドを実行する
目的		アプリケーションから動的プログラム内の関数及びメソッドを実行する
アクター		アプリケーション
事前条件		動的プログラムがインストールされている
事後条件		
フロー	STEP	アクション
基本フロー	1	動的プログラムの初期化をする
	2	動的プログラムファイルの解析をする
	3	関数及びメソッドを実行をする
	4	動的プログラムの終了処理をする
代替フロー		
例外フロー		

ユースケース名		複数の関数及びメソッドを同時に実行する
目的		アプリケーションから動的プログラムのスレッド機能呼び出す
アクター		アプリケーション
事前条件		動的プログラムがインストールされている
事後条件		
フロー	STEP	アクション
基本フロー	1	動的プログラムの初期化をする
	2	複数のスレッドを生成する
	3	複数のスレッドを実行する
	4	複数のスレッド終了待ちをする
	5	動的プログラムの終了処理をする
代替フロー		
例外フロー		

### 3. ライブラリ非機能の要件

#### (ア) システム方式要件

##### ① システム構成の方針

- ・ソフトウェア製品の活用方式  
広く流通し、利用実績のあるものを活用する

##### ② システム全体構成

OSの種類：Vine linux

バージョン： 6.5

アプリソフトの種類：Vine linux6.5 で動作するアプリケーション

#### (イ) テスト要件

- ① システムの利用開始前に十分なテスト期間を確保し、信頼性の確認を行うこと。
- ② システム利用開始後であっても、テスト不足を認められる場合には、必要なテストを実施すること。また、その結果、システムが本業務仕様に適用しない事実が発見された時は、速やかに、見直しを行うこと。

#### 4. ライブラリ詳細設計

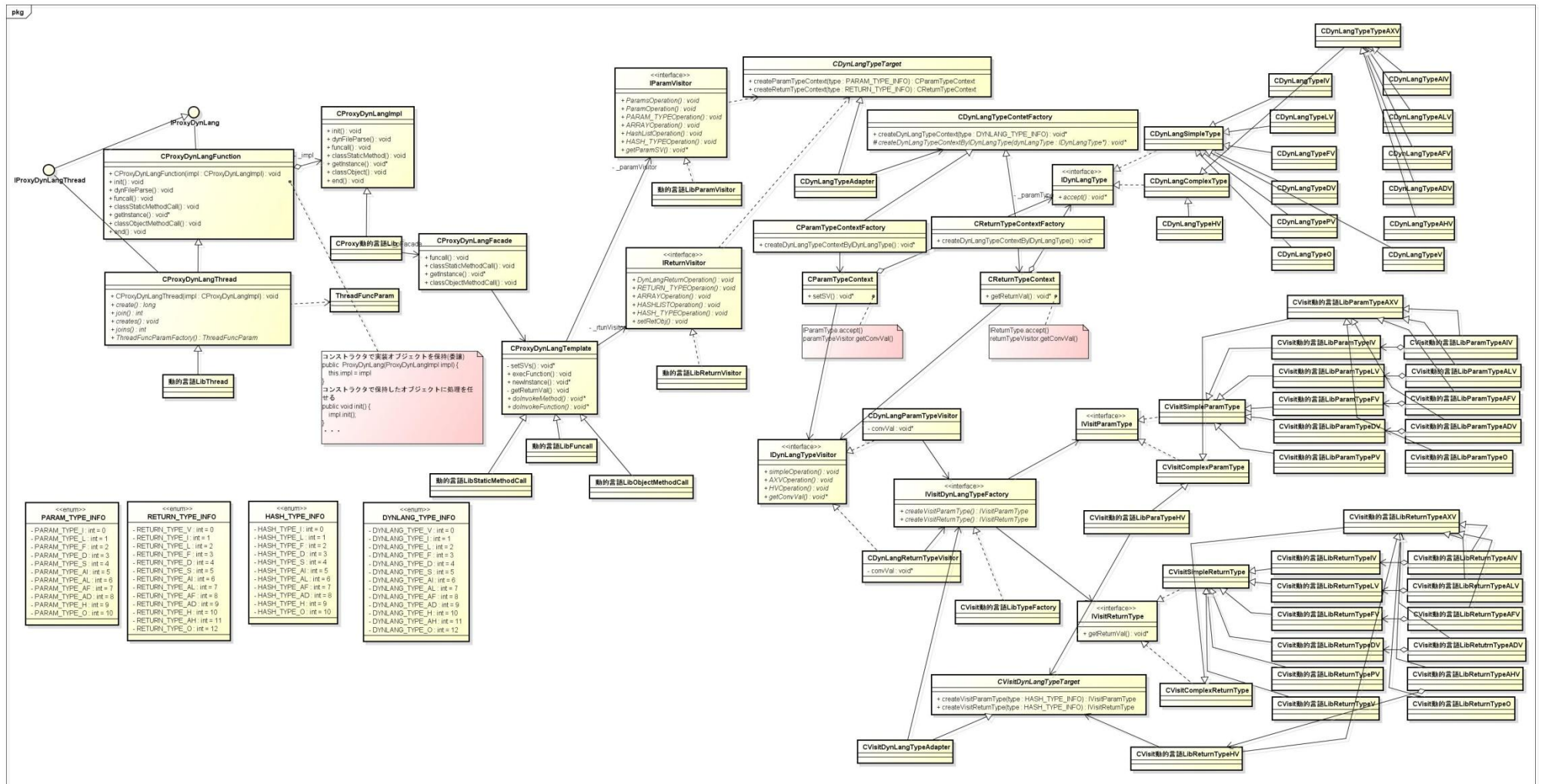
関数及びメソッドを実行機能 USDM		
Req01	要求	アプリケーションから動的プログラムを解析し、関数及びメソッドを実行する
	理由	
	説明	動的プログラム言語のライブラリを利用して、関数及びメソッドを実行する
	<動的プログラムの初期化>	
	Spec-01-1	動的プログラム言語のライブラリを利用して、動的プログラム言語の初期処理をする
	<動的プログラムの解析>	
	Spec-01-2	動的プログラムを解析する
	<関数及びメソッドの実行>	
	要求	Req01-01 動的プログラム内の関数及びメソッドを実行する
		理由
		説明 動的プログラム内の関数及びメソッドにパラメータがある場合は、パラメータを設定し実行する。 結果がある場合は、アプリケーションに受け取る。 メソッドの種類は、以下です。 ・スタティックメソッド ・オブジェクトメソッド データ型は、下記(※1)です
		<関数の実行>
		Spec01-01-1 パラメータがある場合は、パラメータを設定する
		Spec01-01-2 関数を実行する
		Spec01-01-3 戻り値がある場合は、結果を受け取る
		<スタティックメソッド>
		Spec01-01-4 パラメータがある場合は、パラメータを設定する
		Spec01-01-5 スタティックメソッドを実行する
		Spec01-01-6 戻り値がある場合は、結果を受け取る
		<オブジェクトメソッド>
		Spec01-01-7 パラメータがある場合は、パラメータを設定する
		Spec01-01-8 クラスのインスタンスを生成する
		Spec01-01-9 オブジェクトメソッドを実行する
		Spec01-01-10 戻り値がある場合は、結果を受け取る
	<動的プログラムの終了>	
	Spec-01-3	動的プログラムの終了処理

同時に複数の関数及びメソッドを実行機能 USDМ		
Req-02	要求	アプリケーションから、同時に複数の関数及びメソッドを実行する
	理由	
	説明	動的プログラムの同時に複数の関数及びメソッドを実行する機能を利用してする
	Spec-02-1	Spec-01-1 を実行する
	Spec-02-2	複数の関数及びメソッドを生成する
	Spec-02-3	実行する
	Spec-02-4	複数の関数及びメソッドの終了待ちをする
	Spec-02-5	全て終了した場合、Spec-01-3 を実行する

※1 データ型

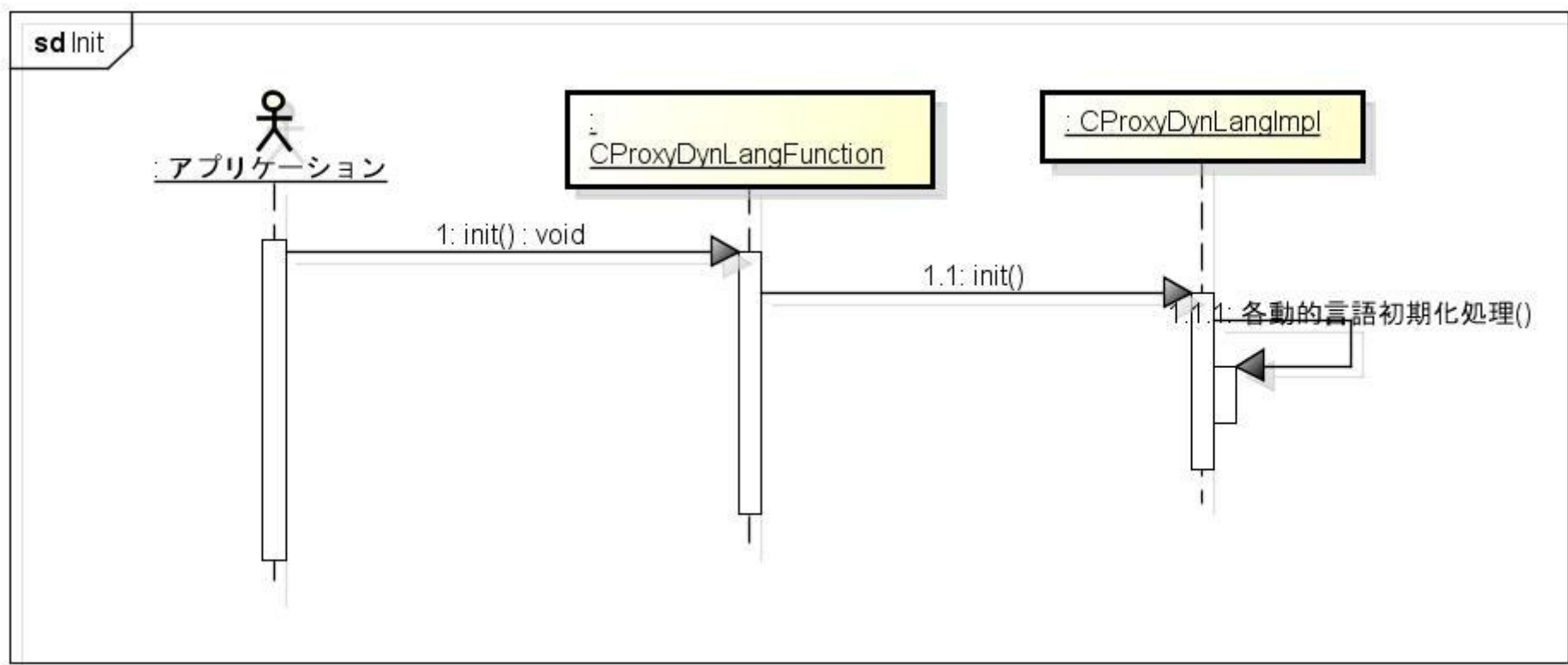
パラメータ値型	ハッシュ値型	戻り値型
		戻り値無し
Int 型	Int 型	Int 型
Long 型	Long 型	Long 型
Float 型	Float 型	Float 型
Double 型	Double 型	Double 型
文字列	文字列	文字列
Int 型の配列	Int 型の配列	Int 型の配列
Long 型の配列	Long 型の配列	Long 型の配列
Float 型の配列	Float 型の配列	Float 型の配列
Double 型の配列	Double 型の配列	Double 型の配列
ハッシュ型	ハッシュ型	ハッシュ型
		ハッシュ型の配列
オブジェクト型	オブジェクト型	オブジェクト型

(ア) クラス図



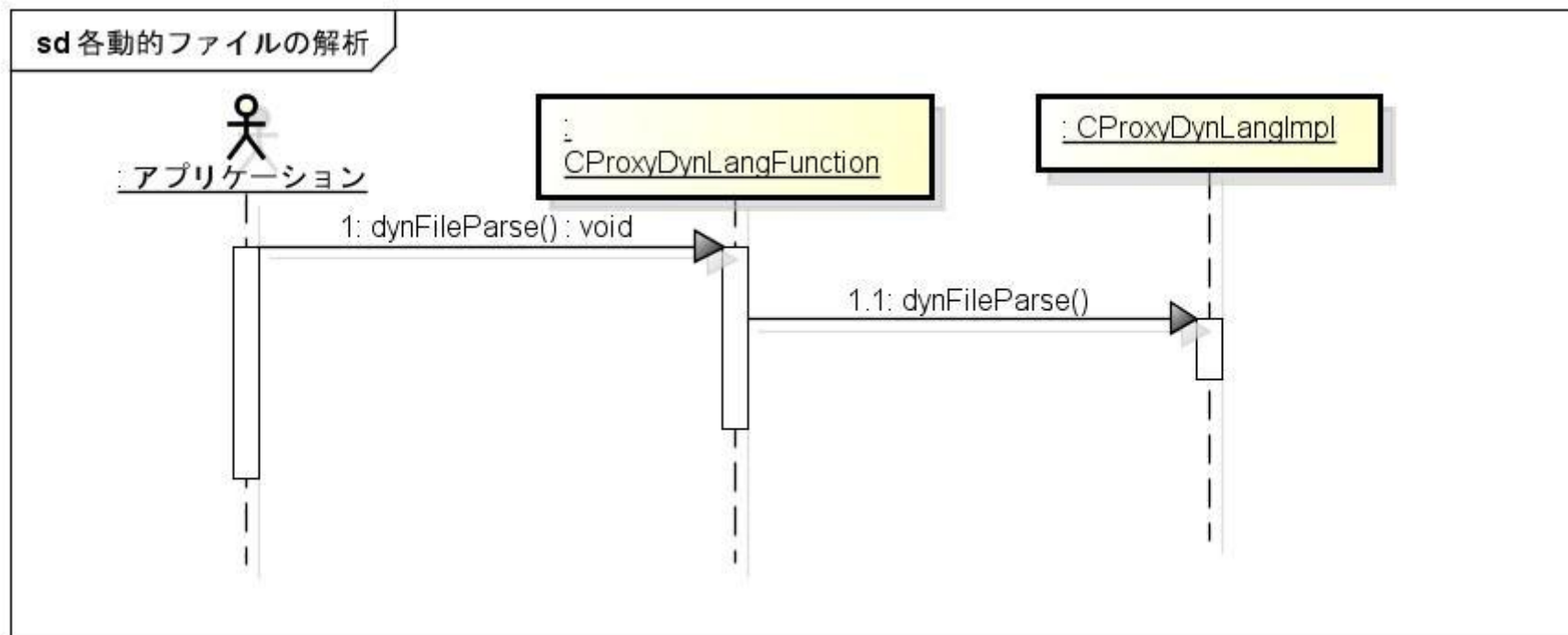
(イ) シーケンス図

① 動的プログラムの初期化

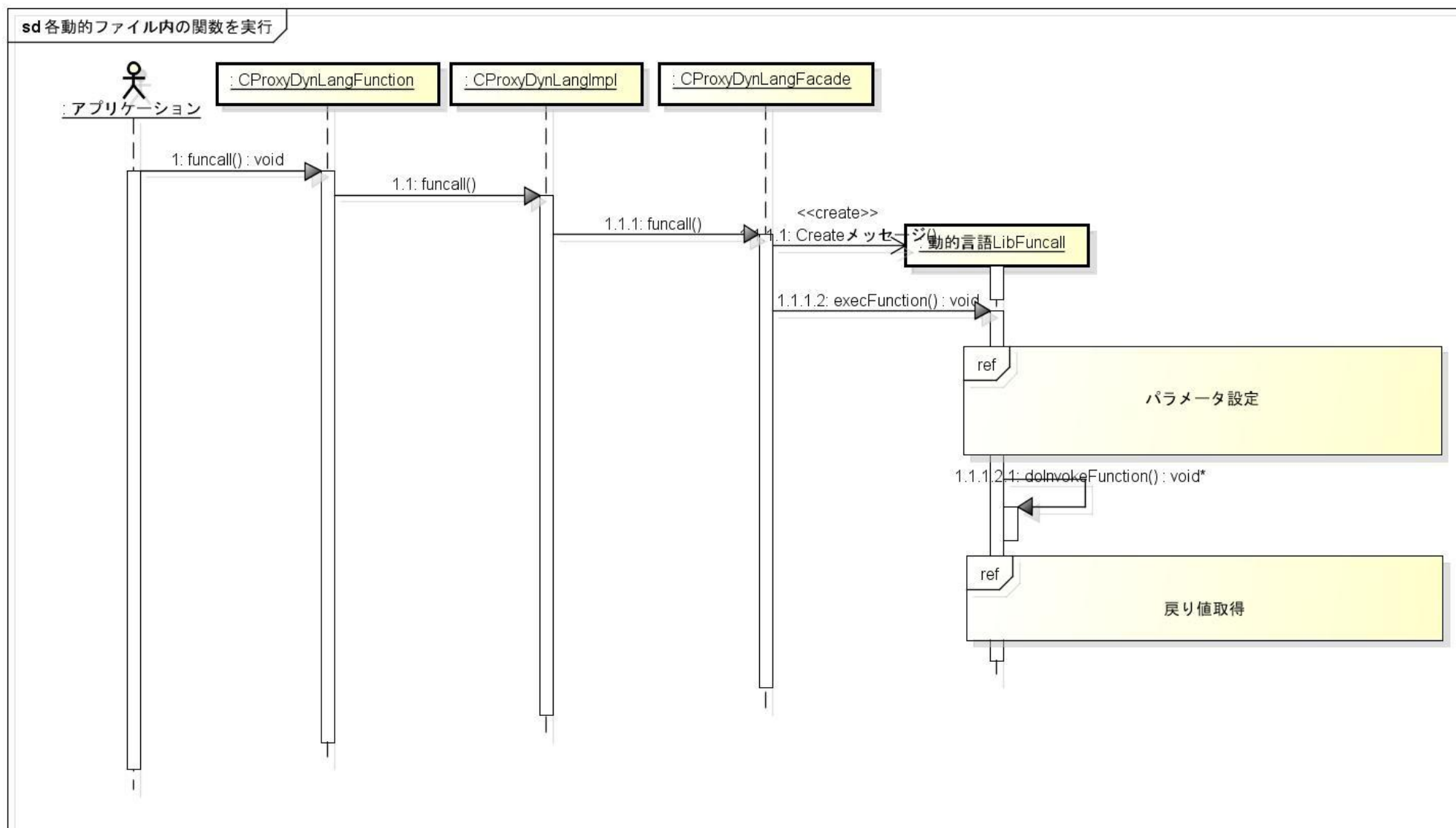




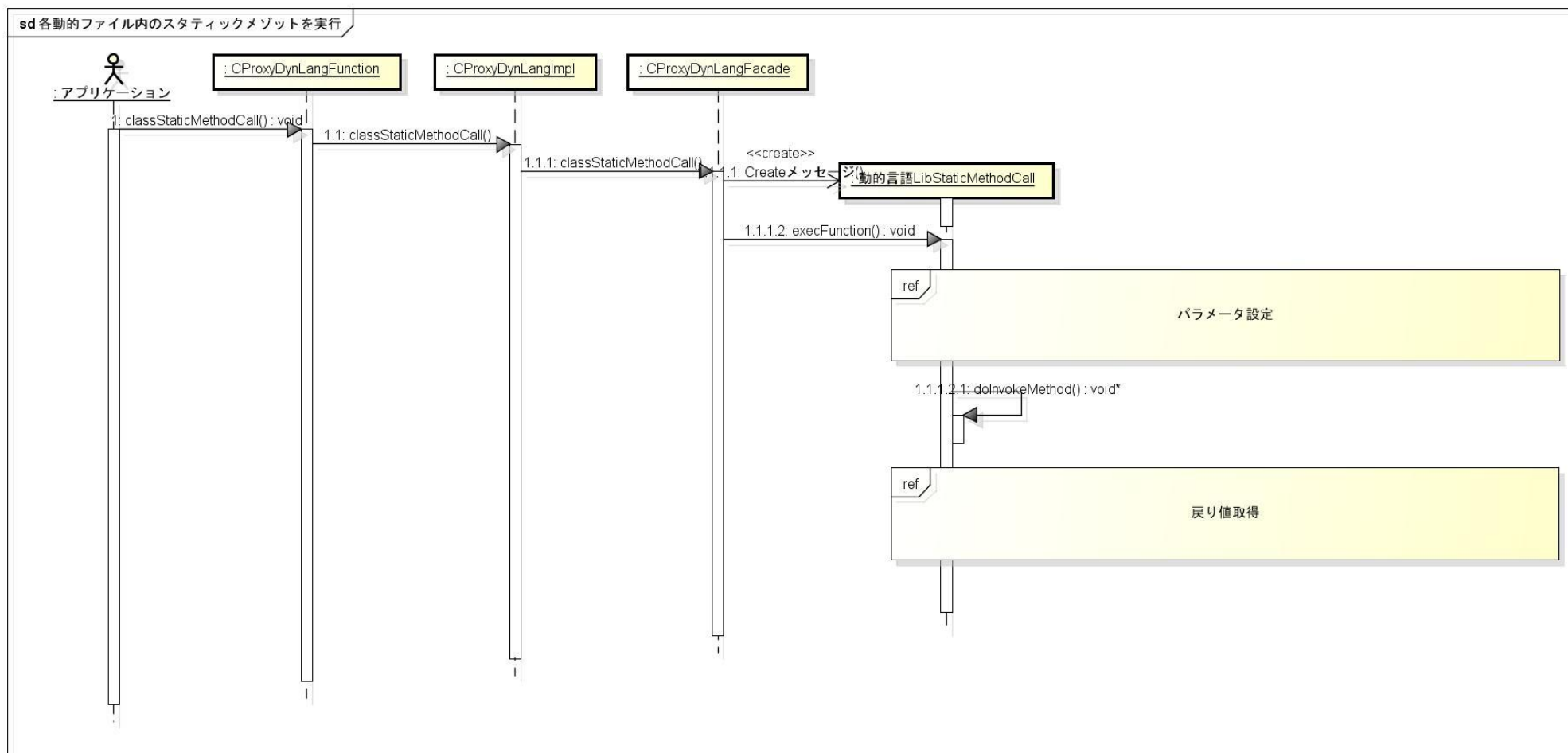
## ② 動的プログラムの解析



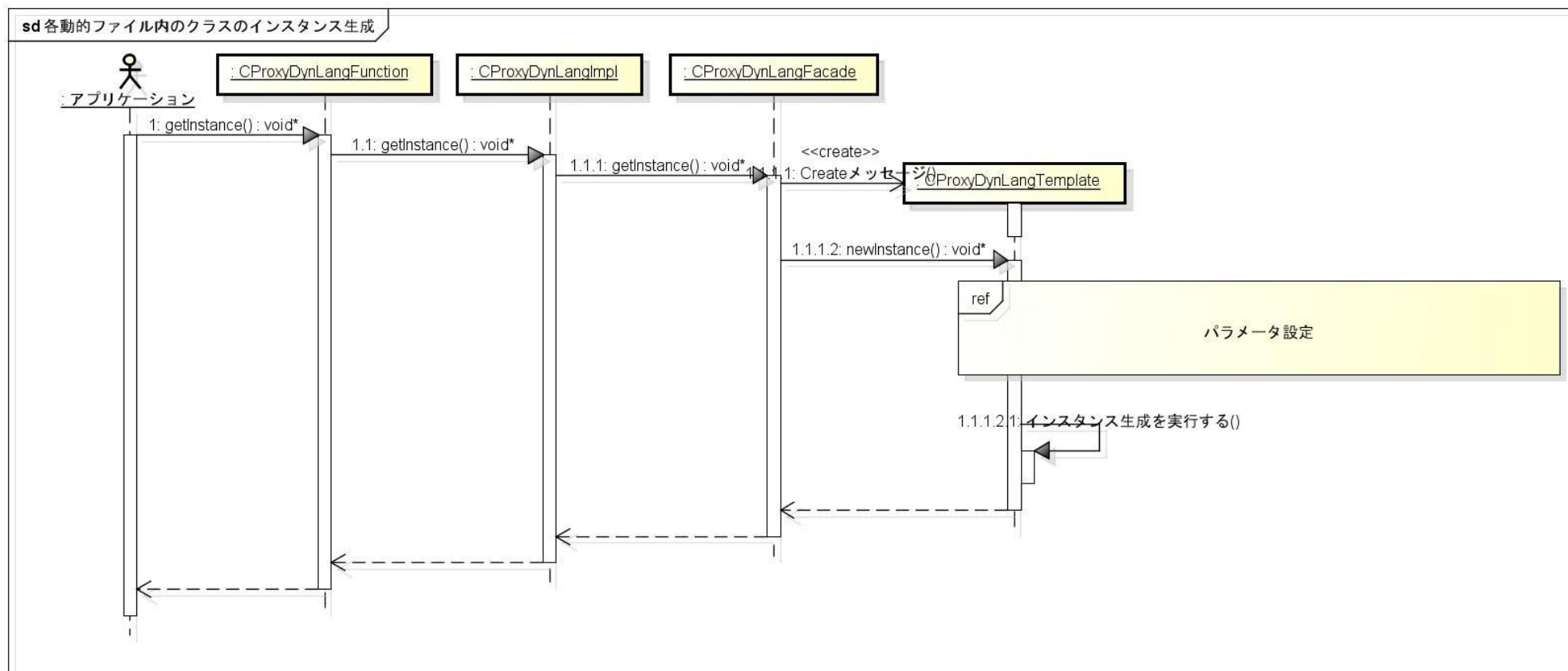
### ③ 動的ファイル内の関数を実行



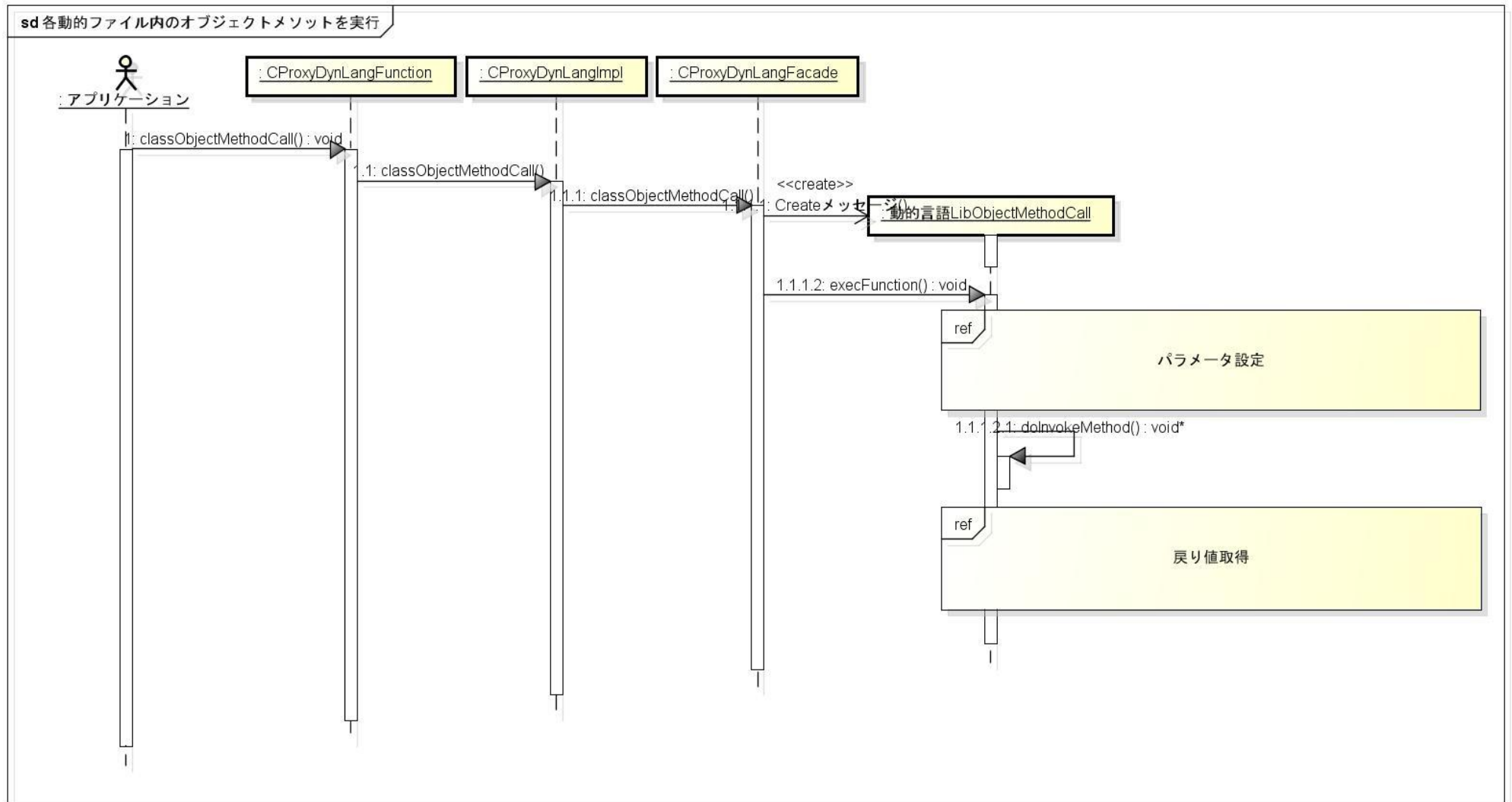
#### ④ 動的ファイル内のスタティックメソッドを実行



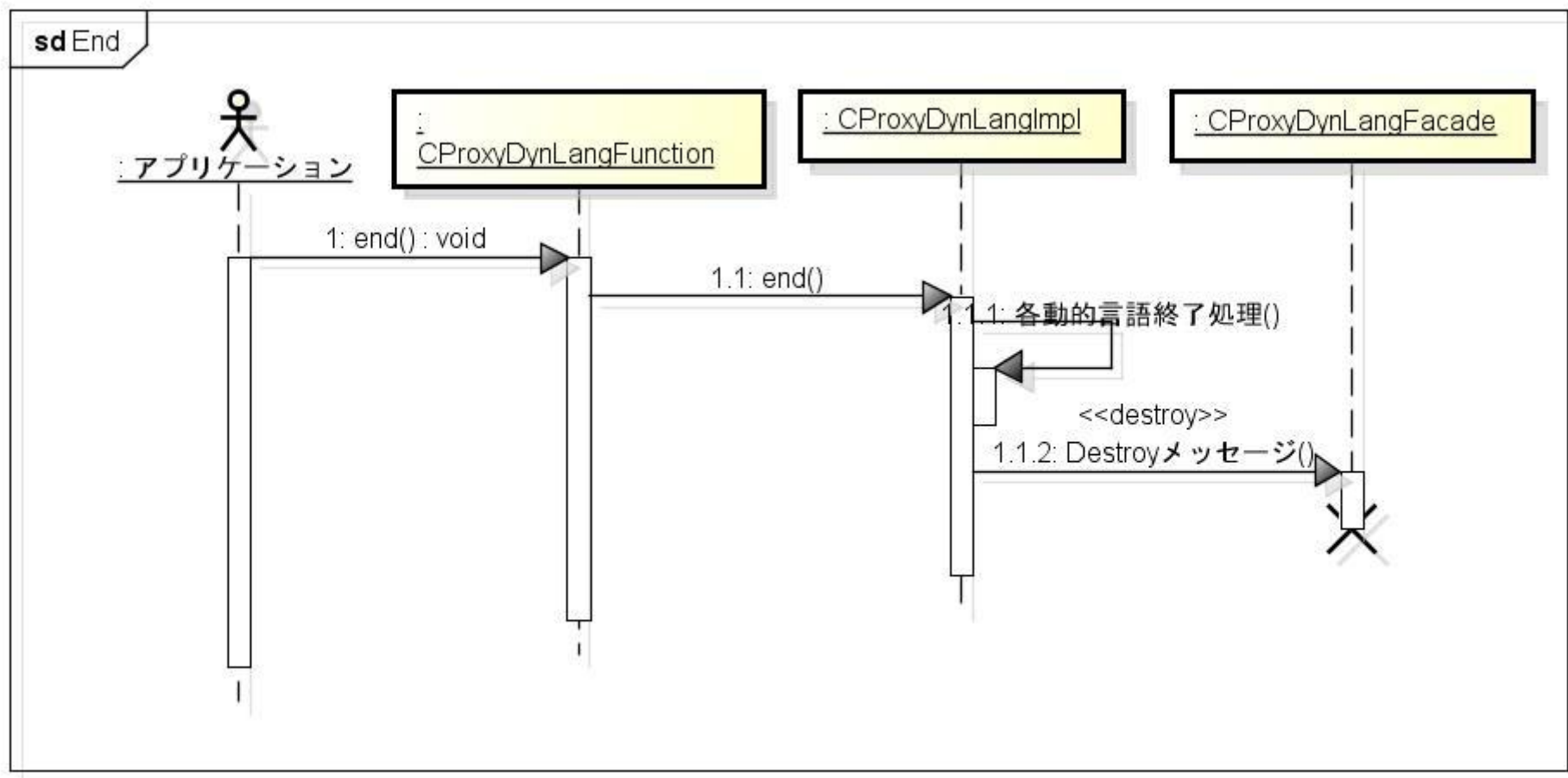
## ⑤ 動的ファイル内のクラスのインスタンス生成



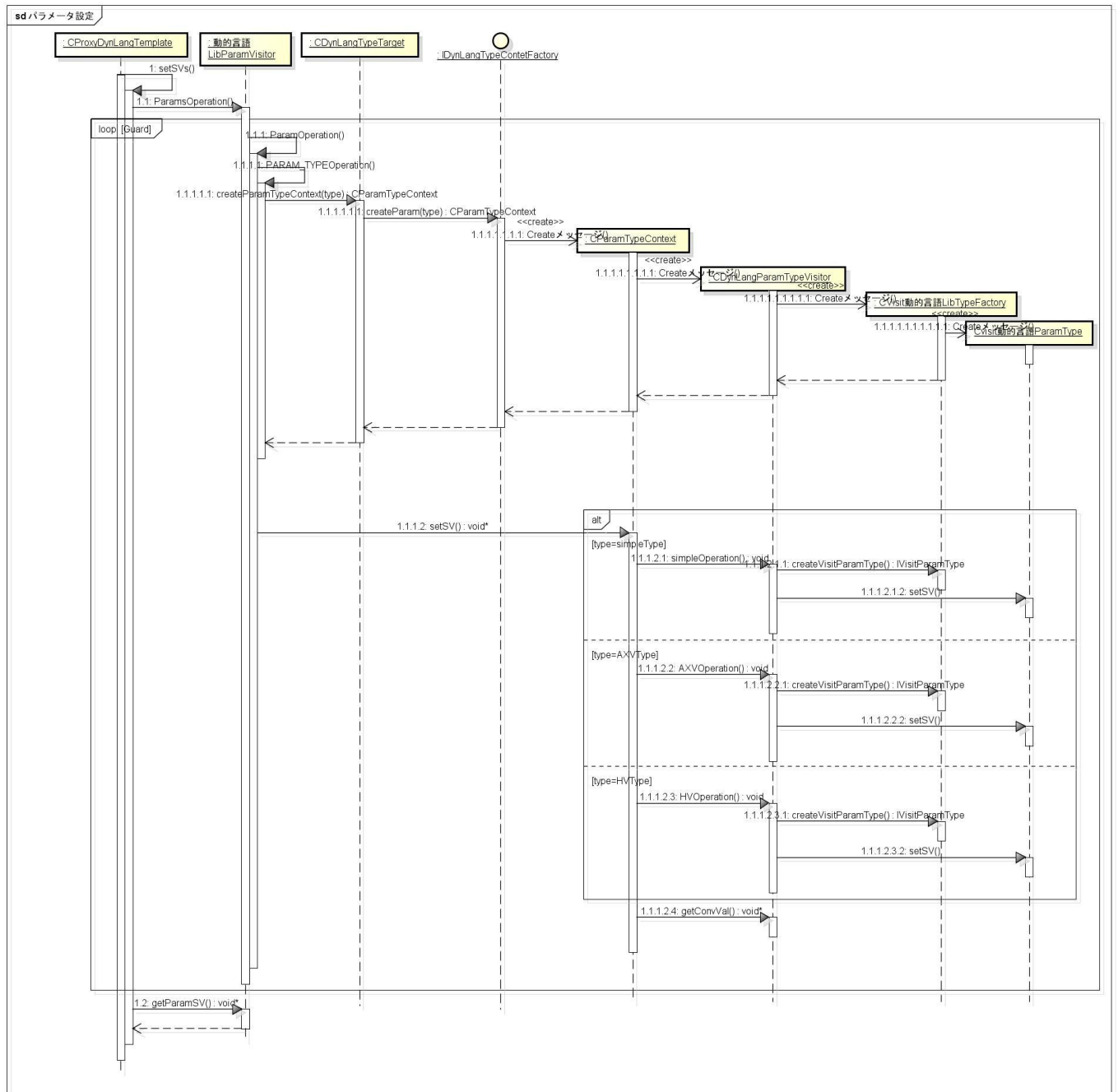
⑥ 動的ファイル内のオブジェクトメソッドを実行



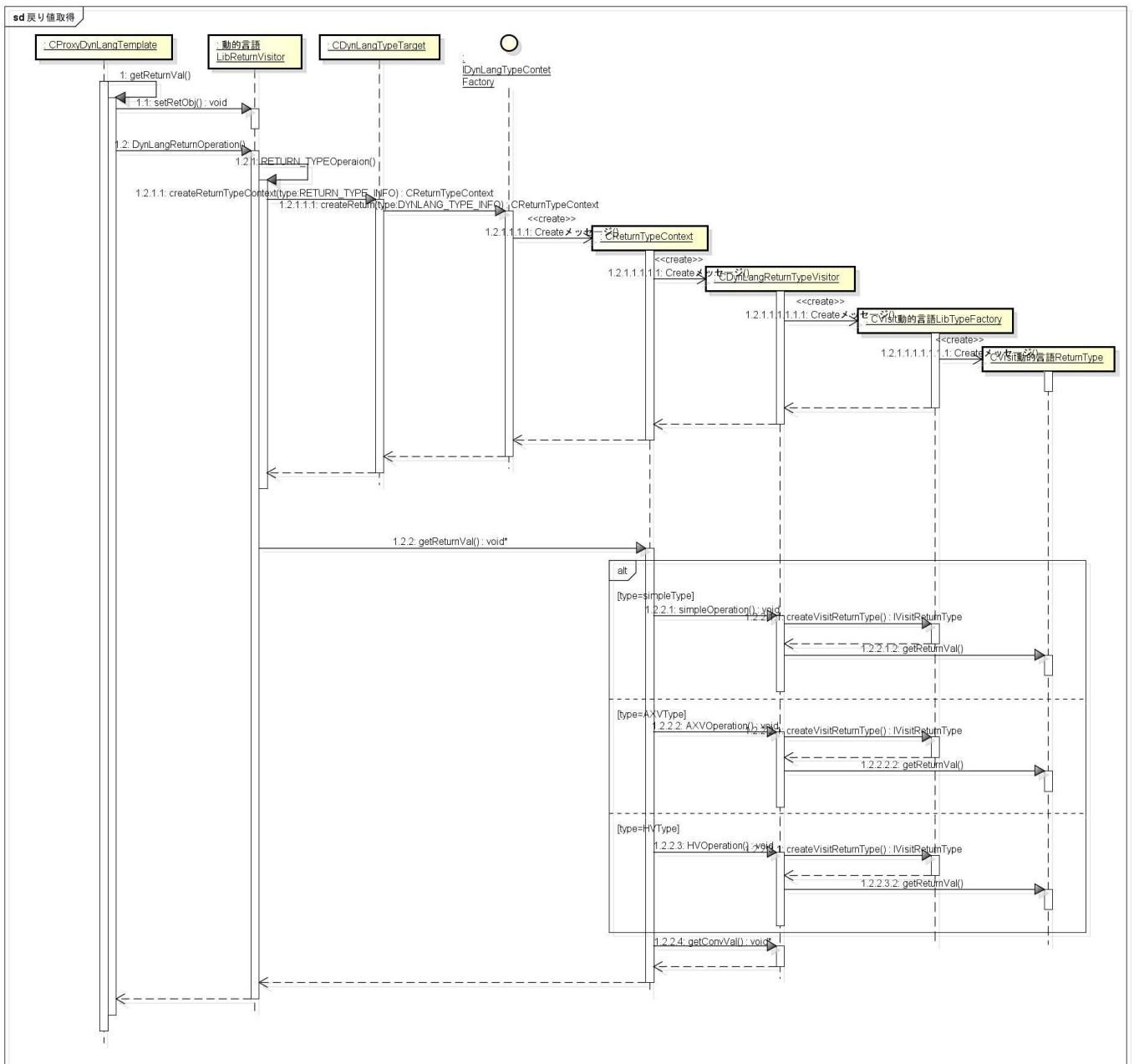
⑦ 動的言語の終了処理



## ⑧ パラメータ設定

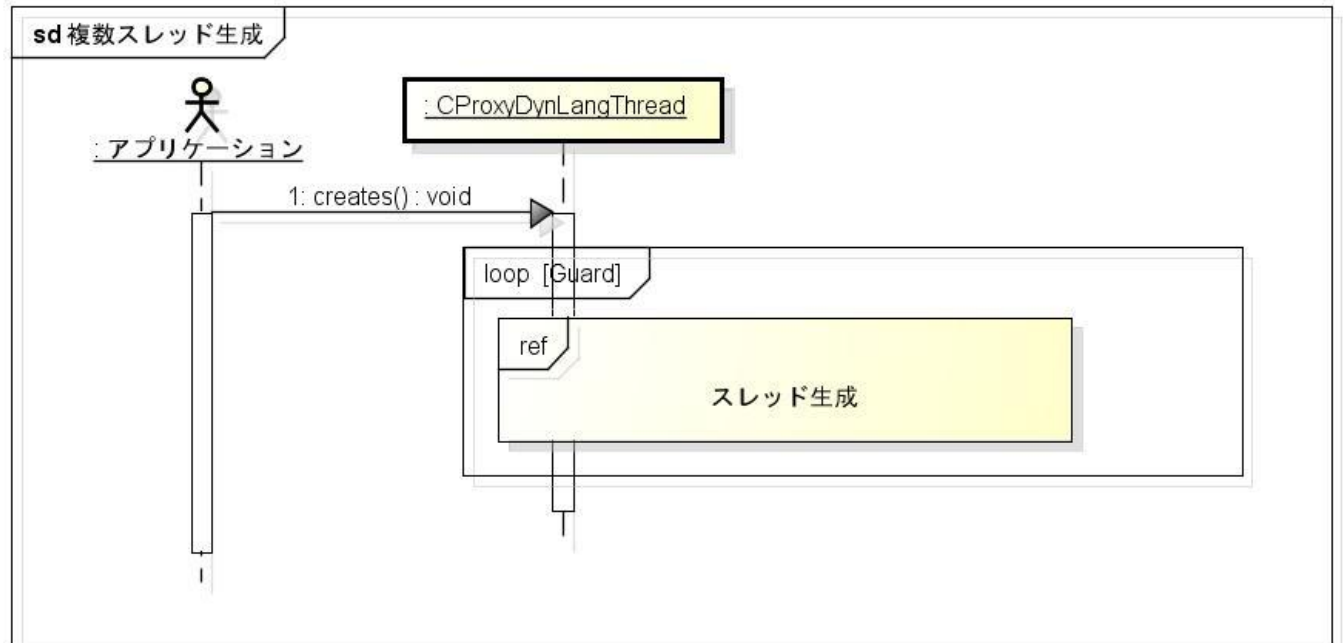


⑨ 戻り値取得



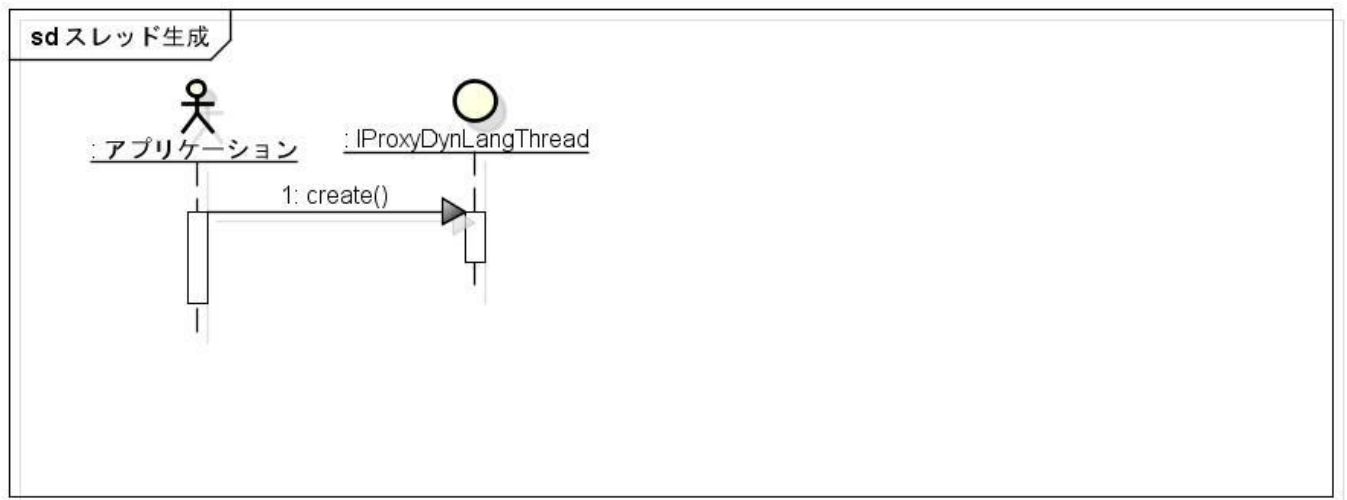


⑩ 複数スレッド生成



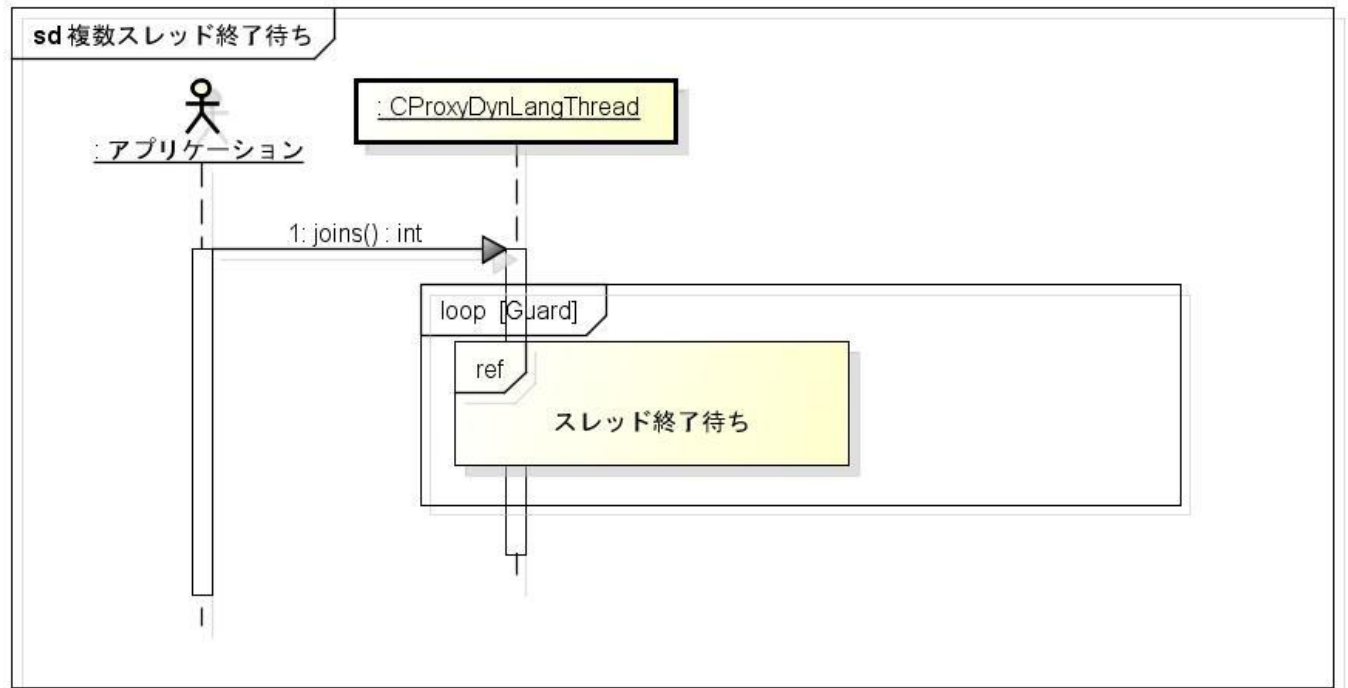
powered by Astah

⑪ スレッド生成



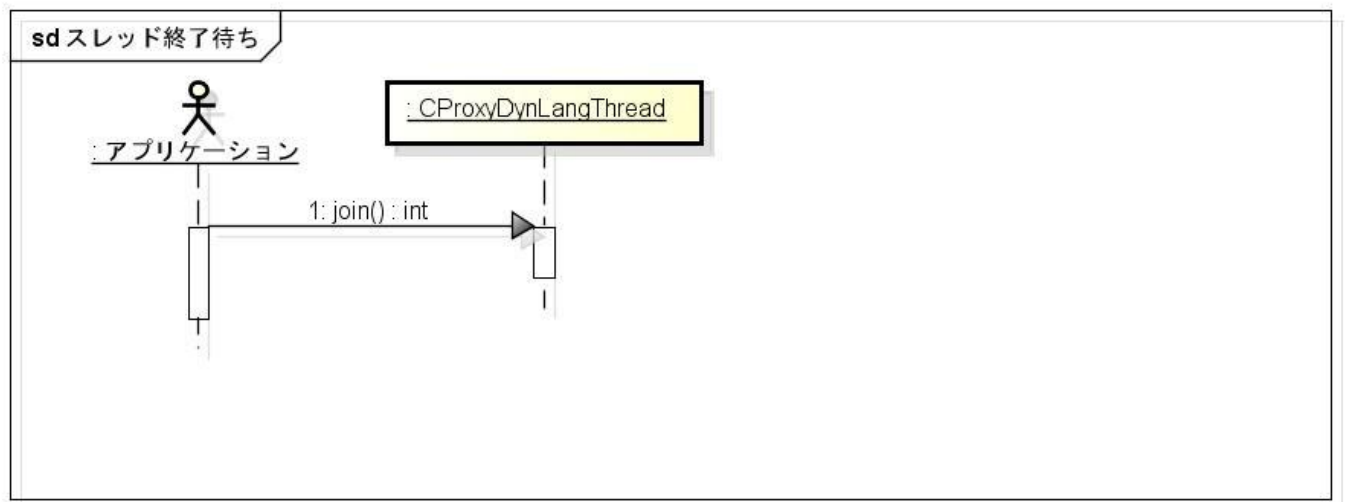
powered by Astah

⑫ 複数スレッド終了待ち



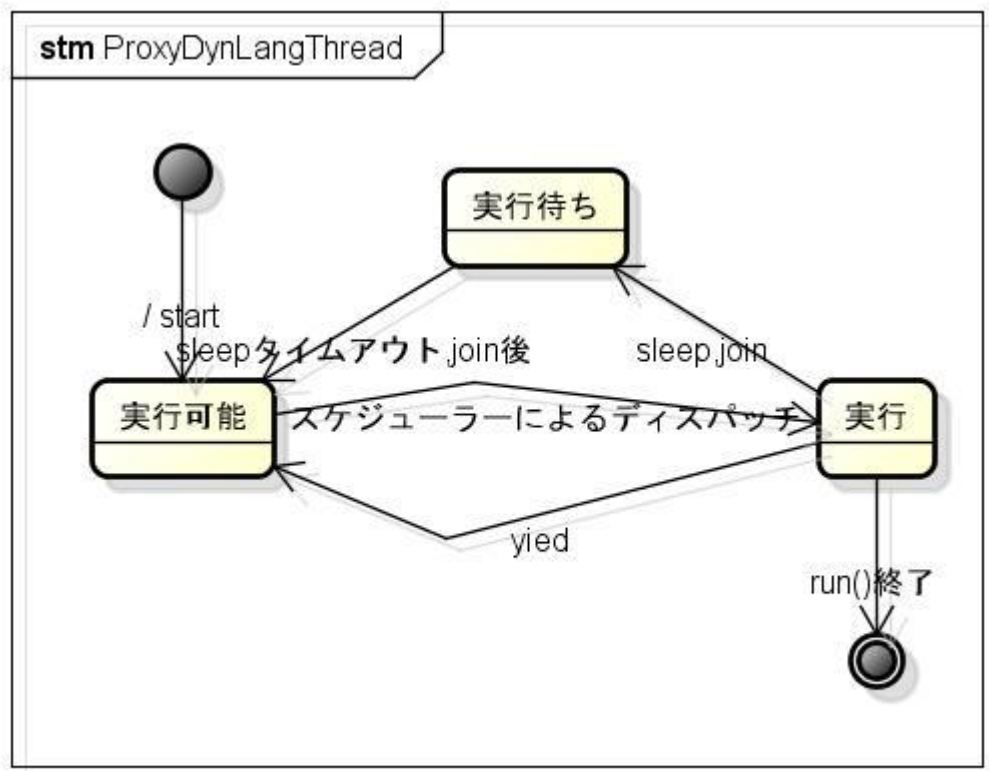
powered by Astah

⑬ スレッド終了待ち



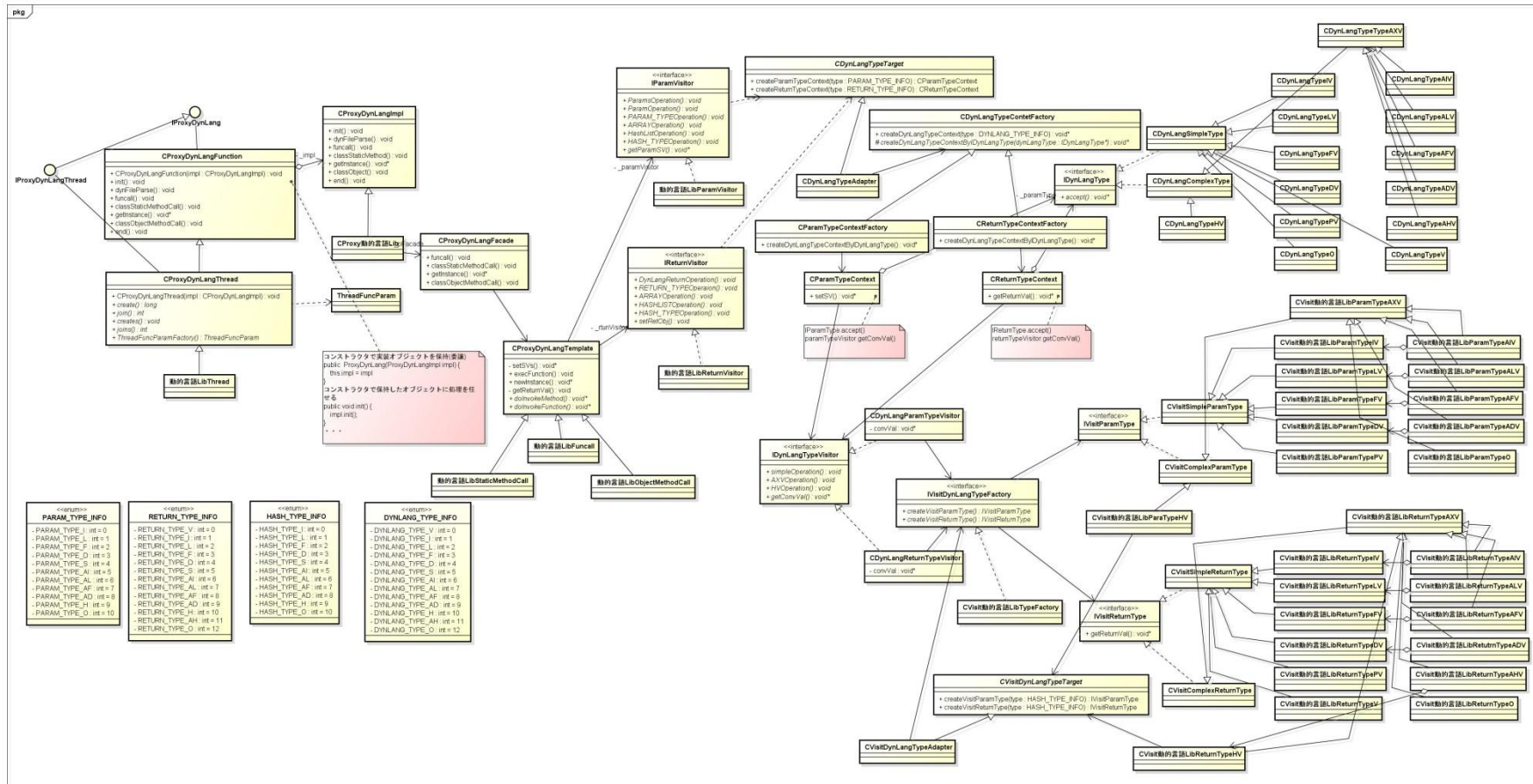
powered by Astah

(ウ) 状態遷移図



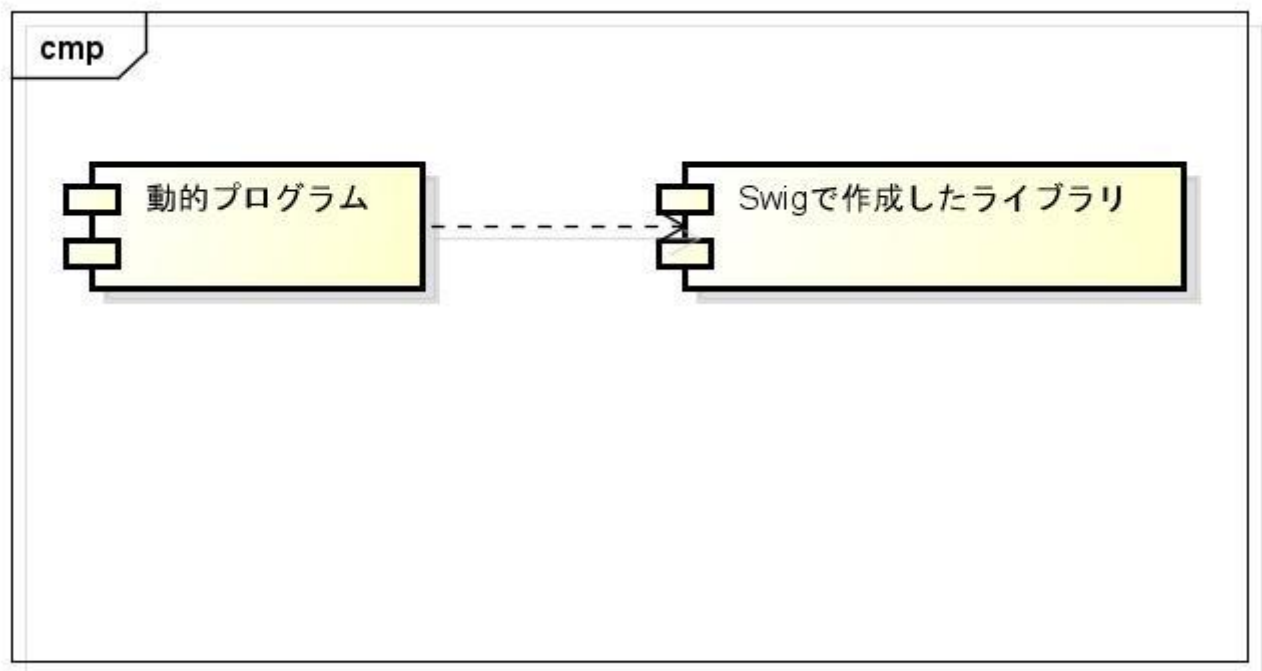


## 5. クラス図



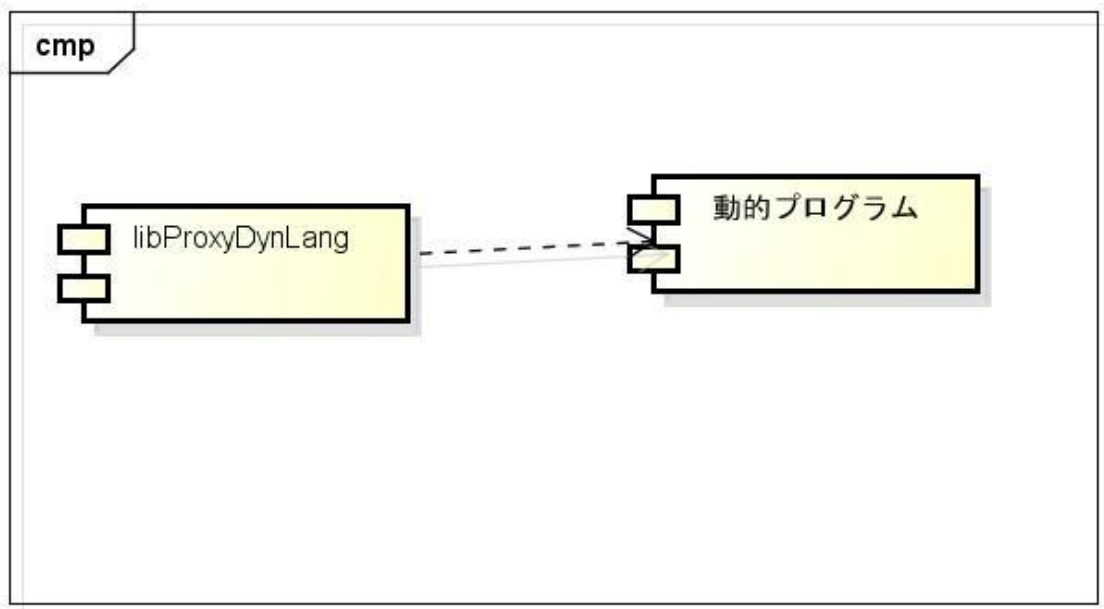
他との対比

- swig



powered by Astah

- 動的プログラム呼び出しライブラリ



powered by Astah