

シーケンス図の階層化

オブジェクトモデリングスペシャリスト

土屋 正人

Masato Tsuchiya

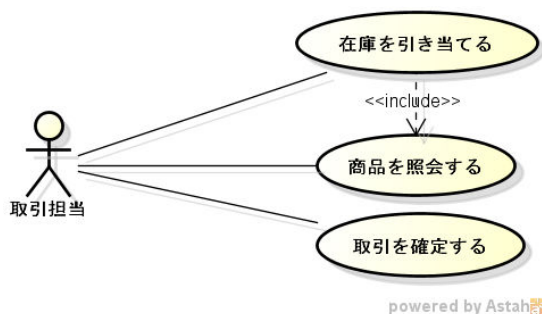
m-tsuchi@sra.co.jp

UML のシーケンス図は、オブジェクト間で行われるメッセージのやり取りを表現するために使われますが、分析から設計に移ってメッセージの粒度が細かくなると、**巨大化して可読性や保守性が落ちます**。A3 用紙に印刷しても、ルーペでも使わないと読めないようなシーケンス図に出会うこともあります。

原因として、対象としているユースケースやストーリーが大きすぎるのが考えられ、その場合はユースケースやストーリーの単位を見直すことで解決できます。とはいえ、妥当な粒度のユースケースやストーリーであっても、設計モデルのシーケンス図になると、巨大化は避けられないことが多いでしょう。

◆“ref”を使う

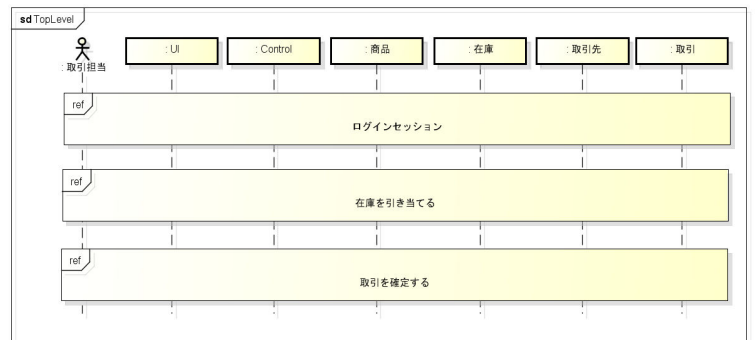
シーケンス図はUML2 大幅に改定されましたが、新たに追加された要素のひとつに、他のシーケンス図を参照する要素“ref”(相互作用の利用)があります。“ref”を使うことで、シーケンス図を構造化することができます。例として、次のような販売業務を考えてみます。



powered by Astah

図 1 ユースケース図

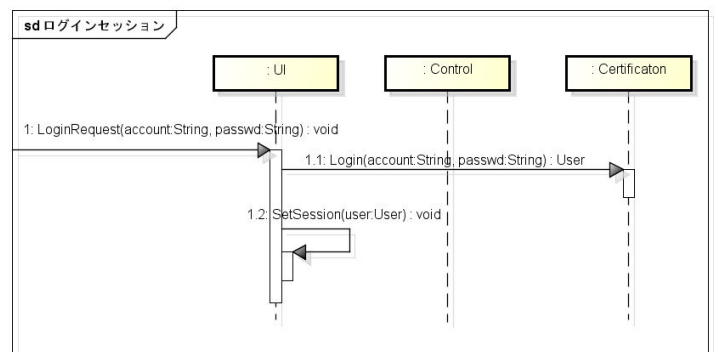
この販売業務のトップレベルのシーケンス図を、“ref”を使って次のように表すことができます。



powered by Astah

図 2 トップレベルのシーケンス図

トップレベルのシーケンス図は、対象とする販売業務では“ref”で示された 3 つの処理単位が実行されることを示しています。“ref”の部分の詳細を把握したいときは、別に作成したシーケンス図を参照することになります。例えば、「ログインセッション」の詳細は次のようになります。



powered by Astah

図 3 「ログインセッション」のシーケンス図

同じく「在庫を引き当てる」の詳細は、次のシーケンス図を参照します。

◆パターンを”ref”で定義する

図 5 では、さらに”ref”を使っています。データベースアクセス部分の詳細を把握するには、「RDB 取得パターン」のシーケンス図を参照します。

データベースアクセスのような汎用的なメカニズム要素は、繰り返し使われるため、対象とするエンティティとそのリストなどを抽象化してパターンにします。個々のユースケースに特化したシーケンス図ではなく、パターンのシーケンス図を作成するわけです。フレームワークを利用する場合は、利用手順を示すことになります。

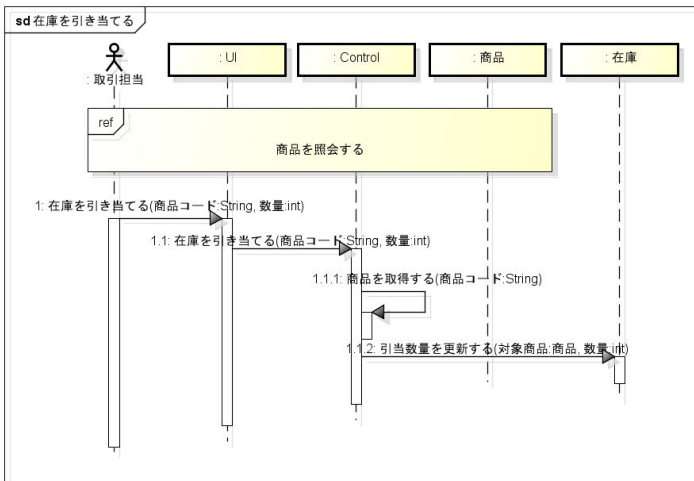


図 4 「在庫を引き当てる」のシーケンス図

このシーケンス図の中でも、「商品を照会する」を”ref”しています。ユースケース図(図 1)で、ユースケース「在庫を引き当てる」がユースケース「商品を照会する」を<<include>>している部分に該当します。このようにして”ref”を使って階層化したシーケンス図として表現することが出来ます。

参照先である「商品を照会する」のシーケンス図は、次のようになります。

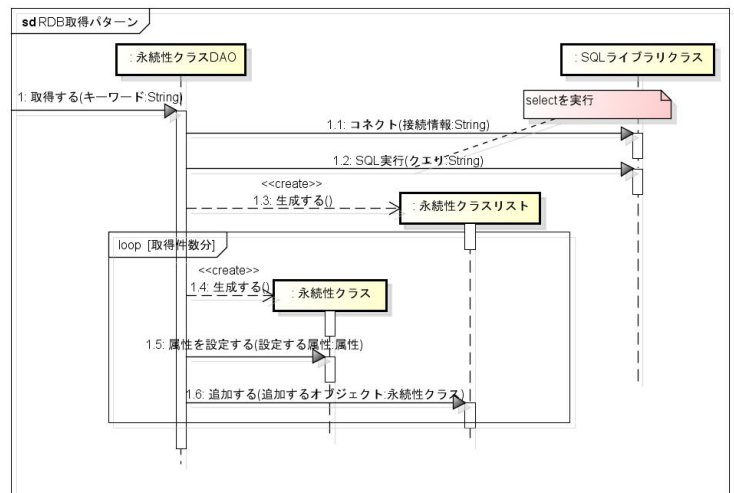


図 6 「RDB 取得パターン」のシーケンス図

登録、更新、削除なども、同様にパターンとしてシーケンス図を作成します。個々のユースケースなどで適用する際には、対象とするエンティティをパラメタとして、抽象化されたライフラインに割り当てます。

図 5「商品を照会する」では、「RDB 取得パターン」の「永続性クラス DAO」に「商品 DAO」を、「永続性クラス」に「商品」を、それぞれ割り当てています。

このように”ref”を使って構造化することで、シーケンス図の可読性、保守性を向上させることが出来ます。

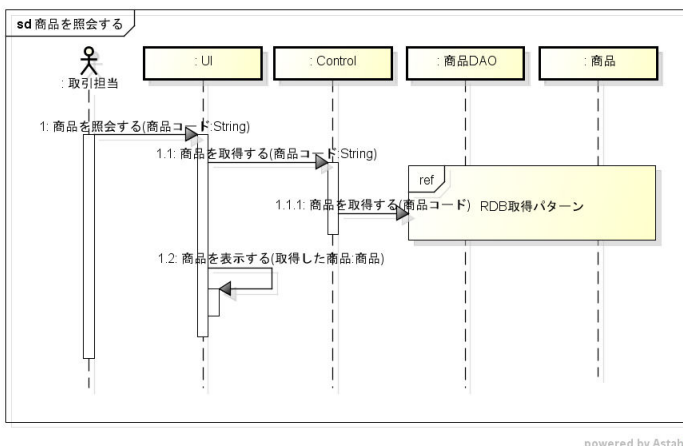


図 5 「商品を照会する」のシーケンス図

GSletterNeo Vol. 60
2013年7月20日発行
発行者●株式会社 SRA 産業第1事業部
編集者●土屋正人、柳田雅子

バックナンバーを公開しています●<http://www.sra.co.jp/gletter>
ご感想・お問い合わせはこちらへお願いします●gsneo@sra.co.jp

夢を。



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-3-2-8

夢を。Yawaraka Innovation
やわらかいのバージョン