

ユースケースモデルを利用した論理サブシステム分割手法について

2C-9

高橋一裕 廣岡龍哉 北村誠

株式会社NTT データ 技術開発本部

ソフトウェア技術センタ

1. はじめに

近年、大規模システムを取り巻く業務環境の変化に伴い、既存システムに対する多様なユーザ要件が発生してきている。一般に、大規模システムの業務要件は複雑であるため、新たな要件を実現する際、既存システムへの影響範囲を限定することが困難であり、これがバックログが発生する要因となっている。

複雑なシステムを業務環境やユーザ要件の変化に短期間に対応させるためには、新たな要件の追加変更に対する影響範囲を限定したシステム構成が望ましい。すなわち、システムの変更に対する影響がある範囲に収まるようにサブシステム化したシステム構成が良いと考えられる。

サブシステム化をどのような基準で行うかについてはさまざまな議論があるが¹、我々は、人がシステムを利用する際に果たす役割をサブシステム化の単位とするアプローチを考えた。システムに関わりを持つユーザからの新たな要件は、この役割単位に生じると考えられるため、この単位を基にシステムを構築すれば、新たな機能追加が生じたとしても、その影響範囲を限定できると考えたからである。

このような人の役割を分析するにあたり、我々はユースケースモデル²に着目した。ユースケースモデルでは、アクターと呼ばれる役割単位にシステムの提供する機能を記述する。アクターの抽出を適切に行えば、アクター毎の要件が混在しないような論理サブシステム分割が行えるのではないかと考えた。

本論文では、システム分析の段階で得られたユースケースモデルを利用し、アクター単位に論理的なサブシステム単位を抽出する手法について、その検討結果を述べる。

2. ユースケースモデルによるサブシステム分割

我々のユースケースモデルを用いたサブシステム分割の基本的なアプローチは、ユースケースの最終生産物に注目することにより、究極に要件を生み出す真の発生源を明らかにすることにより、その単位にユースケースを分類することである。

一般にユースケースは、アクターに対して意味のある結果を返すような処理の系列である。この「意味のある結果」がいつ、何のために必要なのかを知っているのは、その結果の最終消費者であり、「いつ」、「何のために」が要件になると考えられる。ここで、この「意味のある結果」をそのユースケースにおける最終生産物と呼ぶ。したがって、最終生産物を消費するアクターが要件を発生させる単位と考えられる。

例えば、チケット予約システムにおいて、窓口オペレータが顧客の依頼に対してチケット予約を行う場合を考える(図1)。このとき、チケット予約ユースケースの最終生産物は、発券されたチケットである。チケットを予約するというユースケースは、窓口オペレータに関連しているが、このユースケースの要件は、窓口オペレータではなく、チケットの購入者から発生する。

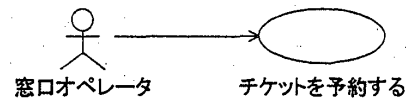


図1 チケット予約システムのユースケースモデル例

要件が発生する単位にユースケースを分類するためには、最終生産物を消費するアクター単位にユースケースを分類すれば良い。これを論理サブシステムの単位とすれば、新たな要件に対して、その影響範囲を限定できるようなサブシステム分割になると考えられる。

3. サブシステム分割の手法

本手法は、サブシステム分割を 1)ユースケースモデルの構築、2)ユースケースモデルの分析、3)論理サブシステム分割という三段階の手順で行う。

3.1. ユースケースモデルの構築

ユースケースモデルを構築する手法は通常通りである。その際、アクターやユースケースの抽出方法が問題となる。

アクターを抽出する方法としては、誰がシステムを使うか、誰が管理するか、他のシステムは何か、決められた時間に自動的に何かが起こるかといった観点でインタビューを実施する。この結果より、システム外部で発生するイベントの発生源を明らかにし、それをアクターとする。

ユースケースは、アクターに対して意味のある結果を返すような粒度に設定する。図 2の図(1)は、初期のユースケースモデルである。ここで、アクターは a_i 、ユースケースは u_i 、ユースケースに関連する最終生産物は fp_i で表す。

3.2. ユースケースモデルの分析

この段階では、論理サブシステム化の観点でユースケースモデルを再構築する。各ユースケースに対して抽出された最終生産物を基準に、それを消費するアクターを関連付ける。図 2の図(2)において、 $a(fp_i)$ は、最終生産物に関して新たに関連付けられたアクターを表す。実際には、この段階で新たなアクターが抽出されたり、異なる最終生産物が一つのアクターに対応する。

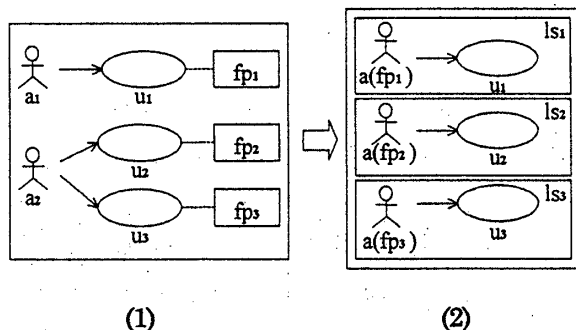


図 2 ユースケースモデルの再構築

3.3. 論理サブシステム分割

ここでは、3.2で得られたユースケースモデルを基に、サブシステム分割を行う。論理サブシステムをアクター単位に定義する。すなわち、図 2の図(2)では、抽出さ

れた各 $a(fp_i)$ について、関連するユースケース集合を論理サブシステム ls_i として定義する。

4. おわりに

本論文では、ユースケースモデルを用いたサブシステム分割の手法として、アクターを論理サブシステムの単位とする手法について述べた。

本手法で構築されたユースケースモデルを利用し、論理サブシステム化を考えることにより、a)新たな要件が発生しても、そのアクターに対応するサブシステム内に変更の影響が閉じるため、メンテナンスが容易、b)各サブシステム内は、アクターに提供する機能だけを実装しているため、他のアクターに対する要件が混在しないといった利点が期待できる。

しかし、ここで抽出された論理サブシステムを、それぞれ物理的に異なる複数のアーキテクチャにそのままマッピングできるとは限らない。使用可能なプラットフォームや性能、信頼性、導入コスト、運用コスト、分割損といったさまざまな要件のトレードオフを考慮し、分割された各論理サブシステムに適合するアーキテクチャへとマッピングする必要がある。

今後は事例適用を通し、アクター毎に抽出された論理サブシステムを、アーキテクチャへマッピングする手法を検討し、本手法の有効性を検証する必要がある。

5. 参考文献

1. Schneider, G. et al. *Applying Use Cases: a Practical Guide*, pp.65-pp.81, Addison Wesley, 1998.
2. Booch, G., Rumbaugh, J., Jacobson, J., *The Unified Modeling Language User Guide*, pp. 233-241, Addison Wesley, 1998.