

要求生成のメカニズムのモデルに関する一考察

中島 毅(三菱電機)、東 基衛(早大)

要求工学プロセスを全体最適の観点から計画/評価するためのシステムアプローチを確立することの重要性を訴え、この目的に適うシミュレーションモデルが構築可能である一例を示す。このシミュレーションモデルは、プロジェクトコンテキストを表現し、要求種別・エラーの構造と依存関係をとらえ、要求工学技法の能力のカタログ化を可能とする。

A Prospect for Systems Approach to Requirements Engineering

Tsuyoshi NAKAJIMA and Motoei AZUMA

The importance of applying the systems approach to planning and evaluating requirements engineering activities is discussed, and the reality of this approach is demonstrated by constructing a simulation model, which is capable of describing project contexts and the dynamism of generating and detecting the requirements errors, and packaging requirements engineering techniques.

1. はじめに

システム開発プロジェクトにおいて、要求変更をなるべく起こさないことがプロジェクト運営において死活問題である。このため、要求の早期確定及び要求変更の制御を行う目的で、ゴール指向/シナリオ指向アプローチ、プロトタイピング、ユーザ参加型開発など種々の要求工学技法が提案されている。これらの技法は、要求工学プロセスの特定の場面で一定の効果があることが認められているが、その開発プロセスに与える影響を定量的に評価した報告は少ない。

要求工学技法を実際のプロジェクトに適用する場合、プロジェクトコンテキスト(プロジェクトを特徴付ける種々のパラメータ)を考慮してそのコスト対効果を把握した上で採用を決めるべきであり、この目的に沿う統一的な基準で技法評価を行うことに対する必要性は高い。

本論文では、要求工学プロセスを全体最適の観点から計画/評価するためにシステムアプローチを確立することの重要性を訴え、この目的に適うシミュレーションモデルが構築可能である一例を示す。さらに、モデルの妥当性検証と予測への応用について議論する。

2. 従来の研究

要求変更が、開発プロジェクトのコスト、スケジュールにどのような影響を与えるかを予測するシミュレーションモデルが提案されている。Demarco[DEM03]は、開発前における5つのリスク要因の評価を行い、モンテカルロシミュレーションを行うことによって、スケジュールとコストの最良見積り点からのずれに関する確率分布を予測している。この見積り法は、既存の見積り法をベースに、5つのプロジェクトコンテキストのみを評価する簡便かつ実用的な方法である。しかし、要求工学プロセ

スの技法の選択が及ぼすダイナミクスは加味していない。

LavazzaとValetto[LAV00]は、変更要求(CR)からスタートする変更プロセスのコストを推定する。彼らは、開発プロセスにおけるプロダクトモデルとプロセスモデルを定義し、それぞれに計測可能な属性を定義し、変更要求に対する影響度関数とコスト関数を出す方式をとっている。基本的に保守時における一つの変更要求のコスト評価を狙っている。この方法は、開発プロセスのダイナミズムを表現しているが、要求源の要求生成のダイナミズムを表現していない。

3. 提案内容

3.1 システムアプローチの重要性

断片的な科学的考察と経験的ガイドラインのみでは、要求工学プロセスの計画立案を行うには不十分である。これは、以下の理由による。

- 要求工学プロセスは、ビジネスの成功という戦略的視点から構成・評価されるべきである。ビジネス成功は、個々の要求工学プロセス（抽出・分析・仕様化・交渉・妥当性確認）の効果と効率の達成を基盤とするが、プロジェクトコンテキストに合わせてこれらの適切な組合せを決定することの方がむしろ支配的な効果を生む。
- 要求源と要求工学プロセスは複雑系を構成する。ビジネス環境、要求エラーの相互作用、ステークホルダと分析者の関係と経験と能力が絡まり、部分最適が必ずしも全体最適につながらない。
- 要求工学プロセスの計画立案では、開発プロセスの効率を最適化する要求品質と出力タイミングの達成のために、顧客との間の有限回数・時間の調整機会を利用する必要がある。

本論文では、要求工学プロセスの研究と知識集積には、シミュレーションを用いたシステムアプローチが必要であることを主張する。これによって以下のことが可能になると考えられる。

- システム開発の全体最適のために、要求工学技法をどこでどのように適用すべきかの指針を得ることができる。
- 経験的データを利用し、より高度な予測に役立てることができる。
- 要求工学における問題構造、因果関係、ダイナミズムへの理解が深まる。分析者への教育的効果を期待できる。

3.2 モデルの表現範囲に関する要件

図1に要求工学プロセスと開発プロセスの関係を示す。

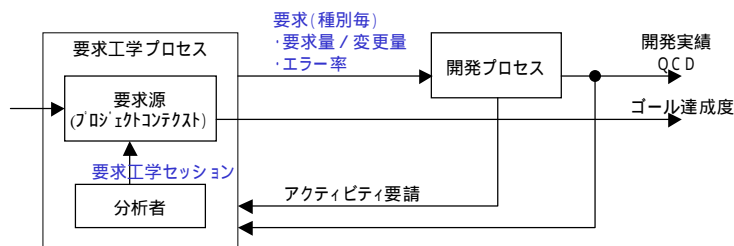


図1 要求工学プロセスと開発プロセス

要求工学プロセスの良否の判定は、単に開発プロセスへの入力である要求の品質を評価すれば

よいのではなく、開発プロセスと組み合わせて閉ループを構成した状態で、ゴール達成度及び品質・コスト・進捗実績を評価する必要がある。

開発プロセスに関するモデルの構成例はいくつか報告されているが[LAV00]、要求工学プロセスのダイナミズムを表現するシミュレーションモデルの提案はない。本論文では、後者に絞りシミュレーションモデルが構成可能であることを示す。モデルは、以下の要件が規定する表現範囲をもつ。

- 要求源のプロジェクトパラメータのみで出力が自動的に決定するのではなく、要求工学セッション投入によって出力が生成されること。この際、作業効果や効率の異なる技法を選択できること。
- ビジネス環境の変化が要求変更を引き起こすメカニズムを表現できること。
- 要求に含まれるエラー種別と、その発生と除去のメカニズムを表現できること。
- 要求は、要求工学プロセスの出力を網羅する要求種別をもつこと。この要求種別は、製品分野の違い及び技法の有効範囲などの差異が現れる程度に分類されていること。

3.2 モデルの実用性要件

システムアプローチを行う上で基盤となる実用性を確保するため、次の要件を追加する。

- プロジェクトコンテキストとして可観測であるパラメータを選ぶこと。
- 要求工学プロセスを制御するためのスケジューリング単位（機会）を設定すること。
- 技法の効果と効率を、計測可能なものに設定すること。
- 個々の要求エラーと要求間のコンフリクトの発生と除去を、要求工学プロセスにおける確率的事象として扱えること。

4. モデル化

ここでは、シミュレーションモデルの具体的なモデル化について説明する。本論文のモデルは、種々の仮定の採用と単純化を経て構成しているため、唯一の正しい解であると主張するものではない。

4.1 要求

(1) 要求種別と種別間の依存関係の扱い

要求には、種別があり、その種別に適合する技法がある。

要求種別としては、ゴール、業務要求（業務プロセス、業務ルール）、システム要求（機能要求、品質要求、外部インタフェース要求、制約）に分けた[AZU04]。なお、この要求種別は固定されたものではなく、要求種別間の依存関係を表現し技法が一まとまりで扱うための要求の単位である。

各種別間には依存関係が設定できる。依存関係を次の2種類に固定した。

作業依存： 依存元の作業の完成度が、依存先の要求抽出の作業効率と、妥当性確認の作業効率とエラー除去率に影響を与える関係。

インタラクション： 要求及び設計の各種別間に、コンフリクトを起こす可能性があることを示す双方向の関係。各インタラクションには、コンフリクトが生じる確率を定義する。基本的には、要求間コンフリクトは抽出作業時に埋め込まれると考える。

表1は、モデルで扱う要求種別と、要求種別間にある特に強い関係を示す。種別において、品質要求はISO9126の定義に沿うものである。設計的決定は、設計に属する事項であるが、設計から要求へ

の強い関係があるため表に加えている。 は作業依存関係、 はインタラクションを指す。作業依存関係はインタラクションを含む。

表1には特に強い依存関係のみを記述したが、シミュレーションモデルでは、すべての要素間で、作業依存とインタラクションをもち、その強さをそれぞれ作業依存度と衝突度で定義する方法をとる。**経験的データの集積により**、各要求間の作業依存度と衝突度の精度を向上させていくことができる。

表1 要求種別と依存関係

		ゴール	業務プロセス	業務ルール	機能要求	合目的性・正確性	相互運用性・標準適合性	セキュリティ	信頼性	使用性	効率性	保守性	移植性	外部インタフェース	制約	設計的决定		
要求	ゴール																	
	業務要求	業務プロセス																
		業務ルール																
	システム要求	機能要求	ふるまい															
			データと変換															
			MMI															
		品質要求	機能性	合目的性・正確性														
				相互運用性・標準適合性														
				セキュリティ														
				信頼性														
			使用性															
			効率性															
			保守性															
	移植性																	
	外部インタフェース要求																	
制約																		
設計	設計的决定																	

(2) 要求工学プロセスと要求エラーの扱い

要求工学プロセスによる要求源への働きかけは**要求工学セッション**に限り、セッションを契機として要求（とその変更）を生成し、洗練していくものとする。**セッション**とは、指定された時刻と時間と参加者で、特定の要求種別に対する特定の作業目的を対象として、選択した技法で行う要求工学の作業単位と定義する。作業目的としては、**抽出、分析、折衝、仕様化、妥当性確認**の5種類を用意している[SWE02]。

シミュレーションモデルでは、**要求エラー**として、あいまい（解釈が複数存在）誤り（顧客が望まない）矛盾（要求間コンフリクト）抜け、実現不能を扱う。図2に、要求エラーと要求エラーを生み出すデータ構造を示す。

要求エラーはすべての要求種別で存在する。要求は、抽出セッションによって生成される。要求が、誤りかの決定は、ドメイン特性と要求種別より定まる**基準誤り率**と、適用技法が指定する**誤り埋込係数(0~1)**により確率的になされる。これは、あいまい/実現不可についても同様である。矛盾は、要求セット間のインタラクションの定義（基準衝突率）と技法の矛盾埋込係数より、要求間に確率的に設定される。分析では、あいまい、抜け、矛盾、実現不能、妥当性確認では、誤りを検知・修正する。エラーの検知も、埋め込みと同様に、**基準エラー検知率**と技法の**検知係数(1以上)**より確率的に処理される。**基準エラー埋込/検知率**および**基準衝突率**は、技法を特に用いない場合の作業効果率として定

義される。これらに関する経験的データの集積が必要になる。

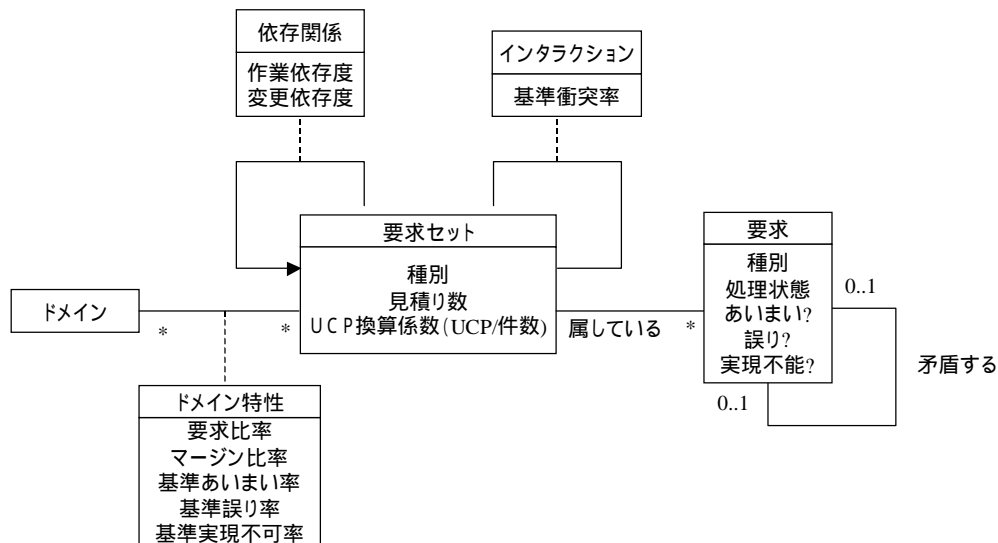


図2 要求と要求エラーのためのデータ構造

(2) 要求総量に対する仮定

このシミュレーションモデルでは、要求種別ごとの要求総量はマージンをもって予め決まっているものとする。折衝は、見積り数を超える場合に要求削減を求めるセッションとして用意している。折衝によって見積り数+マージンの中に収まることになる。これは、見積りが正確で要求工学プロセスが健全なら、本来システム開発予算に応じた規模へ落ち着くよう制御が働くはずと考えるためである。この前提から外れたケースについては、4.4で別途取り上げることにする。

見積りと要求とを結びつける規模尺度としてユースケースポイント(UCP)がある[UCP01]。見積りにおいて、ユースケースがある場合にはそれをカウントし、そうでない場合には予算額から工数単価と生産性係数(20HR/UCP)を使って機能要求に相当するUCPを逆算する。これによって、機能要求に関する見積り総量は、見積りコストあるいは予算と一致したものになる。抽出セッションによる要求生成は、見積り数*(1+マージン比率)まで生成できるものとする。

機能要求以外の要求については、機能要求量(UCP)から見積り量を計算するようにする。機能要求の規模に、ドメイン別の要求比率とUCP換算係数を掛けて見積り数とする。要求比率とUCP換算係数について経験的データの集積が必要である。

(3) 要求工学セッションと技法の扱い

要求工学セッションは、要求工学プロセスの作業単位である。セッションは、どの時点で誰がどれくらいの時間参加し、どの要求セットを対象としてどの技法を用い何を作業するかを定めている。

一つの要求技法は、複数の使用目的に利用することができるため、技法の能力は、仕様目的別にカタログ化しておく。このシミュレーションモデルでは、技法利用によって生じる作業効率の向上、エラー発生量の減少、エラー検知の向上をモデル化できるようにしている。経験的データの蓄積により、要求工学技法のパフォーマンスをカタログ化が必要である。

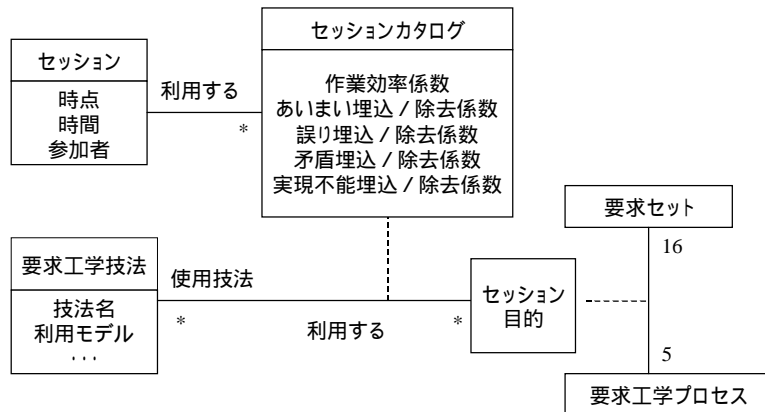


図3 要求工学セッションと技法の関係

4.3 要求工学セッションのダイナミズム

(1) プロジェクトコンテキスト表現

プロジェクトコンテキストは、下記によって決まる。

対象ドメイン

分析者の能力(0 ~ 1)と経験(0 ~ 1)

ステークホルダの数 N と経験(0 ~ 1)、窓口の調整能力(1 ~ N)

要求種別の初期値： 機能改善やリプレース開発の場合には、すでに再利用できるレベルの要求が存在する場合が多い。

これらをプロジェクトコンテキストとして、シミュレーションの初期値として設定する。これらは下記に示すセッションのダイナミズムに係数として影響を与えるようになっている。

(2) セッションのダイナミズム

セッションの作業効率は、適用技法、分析者の能力、ステークホルダの構成、分析者とステークホルダの経験、作業依存する前提要求の作業の進み具合とその影響度から決定される。図4は、抽出セッションにおける抽出量に関する最も単純な決定メカニズムを示している。生成された要求に対して、前述のエラー発生確率から決定したエラー情報が生成された要求の属性データとして保存される。

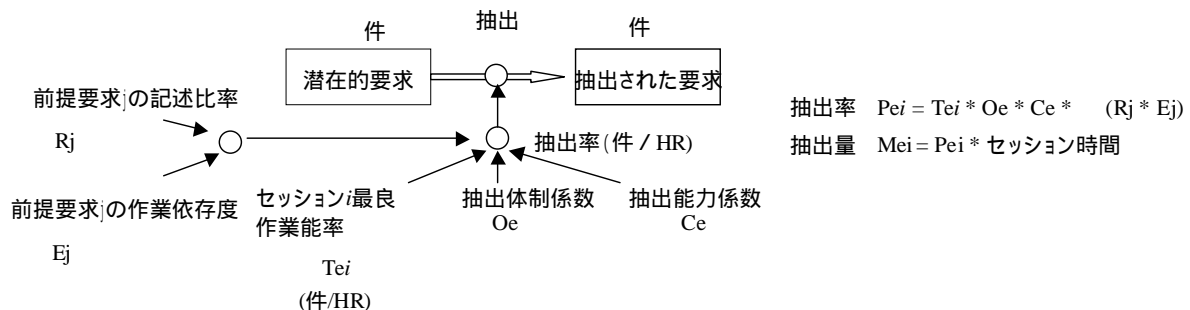


図4 抽出セッションのメカニズム

他のセッションについても、ほぼ同じ作業効率で定義され、さらにエラー除去率のメカニズムが加わるのみである。

5. 妥当性検証とシミュレーションによる予測に関する考察

5.1 妥当性検証

シミュレーションモデルはその正しさを証明することが元々難しい。経験により種々の面から妥当であることを示していく方法しかない。

妥当性検証の一つのアプローチは、比較的小さな単位であるセッションに対して、技法の作業効率（どれだけの要求量を処理できるか）とエラー埋込 / 除去効率の実測を行うことである。これだけのことにしても、ドメインの性質、ステークホルダの性質、分析者の能力、作業依存性をもつ要求の完成度が絡み合うため、効果の適正な評価は難しい。おそらく分析者の能力の効果を生離するのが、最も困難と考えられる。

妥当性検証のもう一つのアプローチは、実プロジェクトの経験データをもとに初期値（プロジェクトコンテキスト）とイベント（実施したセッション）の設定をし、シミュレーション実行して、実プロジェクトの結果である仕様の完成度と比較するものである。

5.2 シミュレーションによる予測

プロトタイピングは、要求抽出と妥当性確認に効果的な技法であることが知られているが、実装作業を伴うため、準備にコストと時間を要する。そのため、コストと時間をあまり要さない他の要求工学的作業の組合せと、プロトタイピングのトレードオフが必要となる。また、要求工学プロセスと開発プロセスのサイクルを複数回まわす反復型の開発の有用性が叫ばれているが、効果的な計画を立てるためには、どのようなプロジェクトコンテキストでどのような反復を行えば、どの程度有用であるのかの大まかな判断材料の提供が必要である。これらの評価には、さらに開発プロセスのモデル化を必要とする。開発プロセスのモデルは、要求変更の影響を受け、開発プロセスのQCDがどのように変わるかを表現しなければならない。こうした試みは、[LAV00]にあり、これらの研究成果を本モデルと組み合わせていくことで、開発全体としての評価を実施していく予定である。

6. おわりに

要求工学プロセスによる仕様生成メカニズムを作業効率とエラー埋込率 / 除去率の観点から、出力としての要求の構造、入力としての要求工学セッションの能力記述、プロジェクトコンテキストを決定するパラメータと初期値について、一通り説明するモデルを構築した。今後はシミュレーション実験を通じて、妥当性確認とモデルの洗練化、予測への適用性の評価を行っていく。

参考文献

[DEM03] Tom Demarco and Timothy Lister, *Waltzing with Bears*, Dorset House, 2003.

[AZU04] Motoei Azuma, *Applying ISO/IEC 9126-1 Quality Model to Quality Requirements Engineering on Critical Software*, RE04-WS.

[LAV00] Luigi Lavazza and Giuseppe Valetto, *Enhancing Requirements and Change Management through Process Modelling and Measurement*, ICRE2000.

[SWE02] *Guide to the Software Engineering Body of Knowledge*, IEEE Computer Society, 2002.

[UCP01] Geri Schneider and Jason Winters, *Applying Use Cases*, Addison-Wesley, 2001.