類似する既存システムの比較による要求ビューポイントの識別

海谷治彦[†]　久保田卓秀[†]
長田晃[†]　海尻賢二[†]

ソフトウェア全体に関連する特性を分離して管理することは，製品の保守性や移植性を高めることに貢献する．しかし，具体的にそのような特性を識別するのは容易でない．本稿では，ある問題を解決する複数の既存システムの比較を通して，適切なビューポイントを識別する方法を議論する．

Exploring Requirements Viewpoints by Comparing Exiting Similar Systems

Haruhiko Kaiya ,[†] Takahide Kubota ,[†] Akira Osada [†]
and Kenji Kaijiri[†]

In this paper, we discuss how to explore rating schemata for software requirements. We try to find such schemata by comparing the characteristics of similar use cases each of which belongs to the different but similar system.

1. はじめに

ソフトウェアの要求分析時点では想定することが困難な状況変化が開発の後段で発生する．よって，分析段階で妥当な要求定義であっても，後段では見直しが必要となる．例えば，競合他社が特定の機能を高品質化すれば，開発途中でもそれに対抗せざるを得ないであろう．よって，要求分析の結果は，起こりうる変化に対して，可能な限り柔軟に対応できるものでなければならない．

我々は類似する複数の既存システムを比較することで，そのような種類のシステムには，どのようなバリエーションが起こりうるかを推測する手法を開発しつつある[1),2)]．我々の手法では，類似した要求項目間にバリエーションが発生するのは，特定の評価基準によって，それぞれの要求項目の測定結果が異なる評価をされたためであるととらえる．ここでの測定法と評価基準を我々はビューポイントと呼んでいる．そして，要求変化は，あるビューポイントを重視するか否かの変化ととらえている．例えば，テレビ放送を録画記録する機能を考えた場合，「高画質」や「低価格」というビューポイントが考えられる．録画機能にゴースト除去機能を追加するという要求変化があった場合，それは，「低価格」ビューポイントを重視するのをやめて，「高画質」ビューポイントを重視するようになったとみなすことができる．別の例として，「低い開発工数」もビューポイントの一種とみなすことができ，ユースケースポイント法を用いれば，ある程度系統的に測定値を得られる．

また，あるビューポイントは全て要求項目に影響するわけではないため，ビューポイントに影響を受けるか否かによって，要求項目を分類することが可能となる．これによって，特定の要求変化に従い，個々の要求項目をどのように変化させるべきなのかの指針を与えることができる．

2. ビューポイントの識別

現状の手法[1),2)]では，類似する既存システムを複数収集し，システム毎にユースケース図を記述する．そして，ユースケースを機能単位とみなし，異なるシステムに属する類似したユースケースを比較する．類似ユースケース間の差異を特徴付ける変数を識別することで，変化の特徴を考察したり，同じビューポイントに影響を受けるユースケースを識別したりする．例えば，ユースケース X とユースケース Y が共に整数変数α の変化に影響を受ける場合を考える．α で特徴付けられるビューポイントの重要度が変化した場合，X と Y 双方についてバリエーションの再検討が必要となると推測できる．さらに，ビューポイント間の関係を整理しやすくするために，ビューポイントを非機能項目[3)]に従い分類する．

† 信州大学 工学部 情報工学科
Faculty of Engineering, Shinshu University
http://www.cs.shinshu-u.ac.jp/~kaiya/

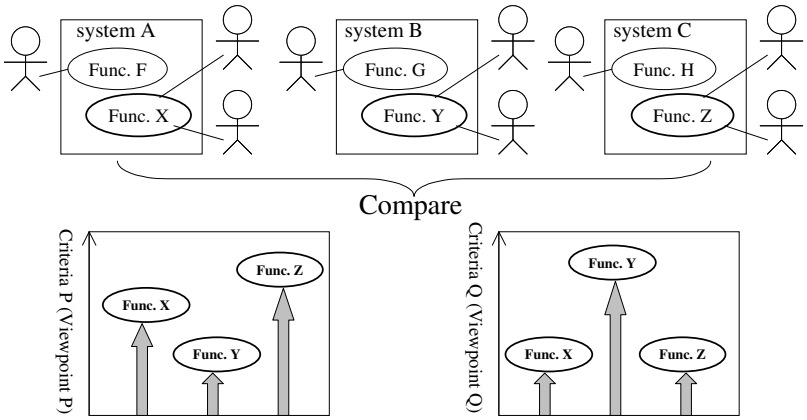

図 1　ビューポイントに基づく類似ユースケース群の順位付け

本手法に基づきケーススタディおよび支援ツールの設計を行う過程で，上記の変数を識別することが非常に困難であることがわかった．そこで現状では図 1 に示すように，異なるシステムに属する類似ユースケースを，個々のビューポイントにおいて，順位付けする手法を検討している．図 1 では，3 つの類似システム A, B, C を比較し，その中にある機能 X, Y, Z が類似しているということがわかったとする．あるビューポイント P からは，$Z > X > Y$ の順に好ましいのに対して，ビューポイント Q からは，$Y > X, Z$ の順に好ましいという結果を図示している．もし，図中の機能 F, G, H も同様に相互に類似しており，ビューポイント P の観点から同様の順序付けがされれば，機能 (X,Y,Z) と機能 (F,G,H) はビューポイント P に共に同様の影響を受けると予想できる．

3. 射影的構造化と階層的構造化

前述のビューポイントは，構築しようとしているシステムを射影的に構造化 (projection) する．これによってシステムに横断的に関わる性質を分離して議論することができる．一方，詳細化や特殊化に基づくソフトウェアの記述法では，伝統的に階層的な構造 (hierarchy) を利用する．本研究での枠組みでは，ビューポイントによって射影的に分離された関心事を，ゴール指向分析によって階層的に表現する．すでにビューポイント毎に階層化されたゴールグラフを合成することで，構築されるシステムのバリエーションの範囲を予測する手法を構築しつつある[4]．

4. 議　　論

本稿では，類似する既存システムの比較を通して，システムに対する要求のビューポイントを識別するための手法を議論した．本手法によって，ISO/IEC 9126 で述べれているソフトウェア製品品質の中でも，特に保守性と移植性を高めることが可能ではないかと思われる．

組込みソフトウェア等，多数の競合他者 (社) との競争の中で製品を市場に送り出さなければならない分野の場合，開発途中に優先するビューポイントの見直しを行わざるを得ないことがある．そして，そのような見直し作業では，迅速に影響範囲を特定し，仕切りなおしを行う必要がある．本稿で提案する手法は，そのような作業の支援に貢献するものと思われる．

本手法は従来は存在しなかった全く新しい機能に基づく変化には対応できない．また，アクタとのインタラクションに関わらないような機能差異は扱い難い．組込みソフトウェアの分野では，IC チップ等のハードウェア部品もアクターとみなすことで，ある程度対応可能だと考えられる．実際，組込みソフトウェアではチップの変更は要求変更として頻繁に起こる例の 1 つである．また，ビューポイントによって比較対象となる機能の集合が異なる場合が考えられる．この点については，さらに検討を重ねてゆきたい．

参　考　文　献

1) 海谷, 長田, 海尻: 既存システムのユースケース図を利用したステークホルダと非機能要求の獲得法, 信学技報, Vol. 104, No. 243, pp. 1–6 (2004). SS2004-14.
2) Kaiya, Osada and Kaijiri: Identifying Stakeholders and Their Preferences about NFR by Comparing Use Case Diagrams of Several Existing Systems, RE'04, pp. 112–121 (2004).
3) Chung, L. et al.: *Non-functional Requirements in Software Engineering*, Kluwer (2000).
4) Kaiya and Saeki: Weaving Multiple Viewpoint Specifications in Goal Oriented Requirements Analysis, APSEC2004, pp. 418–427 (2004).