

パターンの構造化を利用したパターンランゲージの拡充

中野 聡[†] 角谷 将司[†] 鈴木 翔太[†] 鷺崎 弘宜[†] ****
深澤 良彰[†] 羽生田 栄一^{††} 本橋 正成^{†††} 三上 徹也^{††††}

Extending Pattern Languages employed Structuring Pattern

Toshiyuki NAKANO[†], Masashi KADOYA[†], Shota SUZUKI[†], Hironori

WASHIZAKI[†], Yoshiaki FUKAZAWA[†], Eiichi HANYUDA^{††}, Masanari

MOTOHASHI^{†††}, and Tetsuya MIKAMI^{††††}

[†] 早稲田大学基幹理工学部情報理工学科 Dept. Computer Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo, Japan

^{††} 株式会社豆蔵 Mamezou Co., Ltd.

^{†††} 東京工業大学 Tokyo Institute of Technology.

^{††††} 日本アイ・ビー・エム株式会社 IBM Japan, Ltd.

^{†††††} 国立情報学研究所 GRACE センター National Institute of Informatics, GRACE Center

あらまし 多数のパターンからのパターンランゲージの構築により、体系化されたパターン名を用いて言語として円滑なコミュニケーションが可能となり、思考や情報の共有が容易となる。しかし、パターンの抽出のための従来手法では、関連を伴う多くのパターンを抽出することは困難である。そこで我々は、既知のパターンのモデル化とそれに基づく構造変形を利用して、関連するパターンの候補群を導出する手法を提案する。提案手法を用いることで、既知のパターンと関連するパターンを現実的な時間で導出できる。

キーワード ソフトウェアパターン、パターンランゲージ、HAZOP

1. はじめに

ソフトウェアにおけるパターンランゲージとは、相互に関連を持つソフトウェアパターン（以下、パターン）を要素とし、言語のような構造規則に基づいて体系化したものである。多数のパターンから成るパターンランゲージを参照することで、適切なパターンの適用が容易となるのみならず、言語化されたパターン名をコミュニケーションや意思疎通のツールとして活用することができる[1]。しかしながら、過去の資産や経験からパターンを抽出する従来のパターンマイニング手法では、関連を持つパターンの抽出が困難であるため、パターン間の繋がりは不明瞭なものとなってしまふ。その結果、構築されるパターンランゲージは未成熟なものになってしまう場合がある。

そこで我々は、既知のパターンが有する項目を分かりやすい形式でモデル化（以下、この操作を“構造化”と称する）した上で、モデルの構造を変形させる

ことにより、問題に対して解決手段の異なる新たなパターンを導出する手法を提案する。

本論文で扱う研究課題を以下に示す。

RQ1: 既知のパターンの構造化と構造の変形を通じて新たなパターン候補を導出できるか?

RQ2: 導出パターン候補群において既知のパターンと関連し妥当と判断されるパターンが含まれるか?

RQ3: パターンの構造化と構造の変形を通じたパターンの導出を現実的な時間で達成できるか?

本論文の貢献を以下に示す。

- 構造化を利用したパターン候補導出手法の提案。
- ET ロボットコンテストのためのパターンランゲージ[2]を題材としたパターン導出の実証結果。

2. パターンランゲージ構築の課題

2.1 パターンランゲージ

パターンランゲージにおいて多数のパターンは参照しやすい形式でまとめられており、各パターンを頂点、参照関係を辺とするグラフ構造として捉えられる。パターンランゲージを参照することで、問題発見が支援され、また、特定問題に対する経験則としてのパターンを適用可能となる。さらに、パターン名を既知の問題に対する共通の語彙とすることで、意思疎通や思考、情報の共有が容易となる等のメリットがある。

2.2 パターンランゲージ構築における課題

パターンランゲージの利用価値を向上させるためには、関連を伴う多数のパターンを抽出し、それらを基に構築することが望ましい。パターンマイニングの代表的な手法として、有識者によるワークショップやインタビューを通じて得られた知見をパターンとして抽出する手法が存在する[2]。しかしながら、それらの従来手法は個々のパターンを半構造化された文書のまま扱った上で、参加者自身の経験に基づくパターンを模索するため、周辺の新たなパターンの可能性を特定することは困難な可能性がある。結果、抽出されるパターンは既存のパターンとの繋がりが不明瞭となり、パターンランゲージを構築する上でのパターン間の繋がりの表現も困難となる可能性がある。

パターンの文書の例として、“ET ロボットコンテストのためのパターンランゲージ[3]”に収録されている“まいまい式”という既知のパターンを簡略化した記述を以下の表1に示す。この例では、まいまい式と同一のコンテキスト上でスピードを重視する必要性に着目できれば、スピードが求められる場合の関連パターンを導出できる可能性がある。しかし、これまで

のワークショップにおいてそのようなパターンの検討は明示的にはなされていなかった。

表1 まいまい式 パターンの文書

Table 1 Text of "Maimaishiki Pattern".

名前	まいまい式
コンテキスト	コースを走行するための開発を進めている。
問題	会場には外乱光の強い区間が存在する。
フォース	<ul style="list-style-type: none"> ・外乱光によるコースアウトの心配がある。 ・スピードを重視する必要はない。 ・開発期間にあまり余裕が無い。
ソリューション (解決策)	ライトセンサの明滅を利用することで、外乱光の輝度を取り除くことができる。ただし、ライトセンサを明滅させながらの走行となるため、走行速度は僅かに低下してしまう。

3. 構造化に基づくパターン導出

我々は、既知のパターンをコンテキストとフォースから導かれるソリューションとして構造化した上で、フォースを部分的に変更することにより、導かれるソリューションが異なる新たなパターンを導出する手法を提案する。提案手法は、既知のパターンを構造化する手法と、構造を変形させて新たなパターン候補を導出する手法の2段階から構成され、合わせて既知のパターンへと適用することを意図している。提案手法の全体像を図1に示す。図1中でパターン構造からパターン候補を導出する過程については、人手による精査を通じてパターン候補の選定を行うものとする。

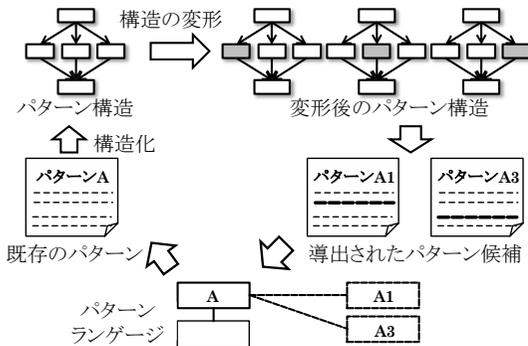


図1 提案手法の全体像

Figure 1 Whole image of our method.

3.1 パターンの構造化

パターンの記述内から特定の要素を抜き出し、有向エッジを用いた階層構造としてパターン構造を表現する。構造化を行う上で、パターンとパターンランゲ

ージ、およびパターンに含まれる項目を整理したメタモデルを図2に示す。

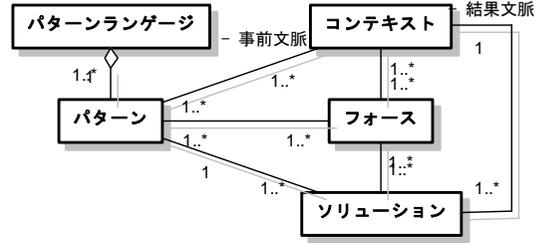


図2 パターンの関連要素を表現するメタモデル

Figure 2 Meta-model describing relevant elements of pattern.

図2においてコンテキストとは、問題が発生する状況について記述した文脈である。フォースとは、コンテキストにおいて問題を発生させる影響力であり、問題解決の上で考慮する事項である。ソリューションとは、フォースの考慮の上で問題を解決する具体的方法である。これらを以下の手順(1)-(4)に従って表1で例示した各項目から導出し、パターン構造を作成する。

- (1) パターンの記述におけるコンテキストや類する項目から、問題が発生する状況と読み取れる事柄を、一つの簡潔なコンテキストとして識別する。
- (2) パターンの記述におけるフォースや類する項目から、問題解決にあたり考慮すべき事項と読み取れる事柄を、簡潔なフォースとして一つ以上識別する。
- (3) パターンの記述における解法や解決策の項目から、問題解決の具体的な方法と読み取れる事柄を、一つの簡潔なソリューションとして識別する。
- (4) 識別したコンテキスト、フォース群、ソリューションを関連付けて、両端の妥当性や、抽象度が同程度であることを確認し、必要であれば修正する。

メタモデルに照らした構造化は、既存パターンの記述を見直し整形する機会となる。例えば“シニアとジュニア”パターン[3]の記述にはフォースとして“人数が複数人いる”が含まれているが、メタモデルの要素に照らすと、これは当該パターンの問題を生じるコンテキストそのものと考えられる。このようにフォースとしては非妥当な記述や、複数の事柄が混在している場合に次節の変化操作を適用すると、本来導出しうる多様なパターン候補群を導出できない可能性や、コンテキスト上ありえないようなパターン候補群を導出してしまいう可能性がある。メタモデルに照らしてフォースその他の記述を整理することで、以降のよ

りの確で詳細な構造変形をしやすくなる。

また、構造化済みのパターンがあれば、それを参照して他のパターンを一貫性のある形で記述しやすくなり、パターンランゲージとして重複なく一貫した形で拡充しやすくなる。例えば、まいまい式の記述にはフォース“スピードを重視する必要はない”が含まれ、関連する“キャリブレーション式”パターン[2]の記述にはフォース“速く走行したい”が含まれる。ここでまいまい式が構造化済みであれば、キャリブレーション式の構造化にあたりフォースの表現を“スピードを重視する必要がある”として捉えることで、その後のフォースの変化操作を一貫した形で実施できる。

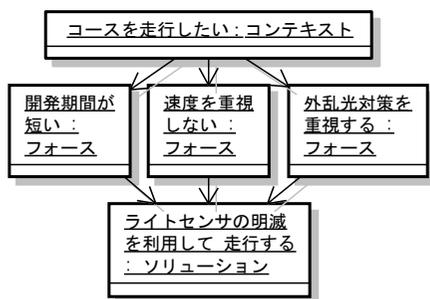


図3 まいまい式パターンを構造化させた例
Figure 3 Example of structuring for “Maimaishiki pattern”.

3.2 パターン構造の変形

パターン構造を変形させるため、フォースに着目する。フォースは、コンテキストにおいて問題を発生させる影響力であり、その考慮によりソリューションを導く働きを担っている[4]。したがって、フォースを変更することにより、コンテキストにおける問題を解決するためのソリューションが既存のものとは異なる新たなパターンの導出が可能となる。

自然言語記述であるフォースに形式的な変更を施すため、HAZOPにおけるガイドワードの考え方を応用する。ガイドワードとは、典型的な形容詞や強調詞、およびそれらの反意語をまとめた語のリストであり、記述の一部分をガイドワードと置き換えることでさまざまな状況を想定する手法がHAZOPである[5]。

HAZOPの手法に基づき、記述の一部に対して“反意語の適用”と“強調詞の付加”の2通りの操作を行うことでフォースの変更を表現する。記述内の形容詞に対して強調詞の付加を、形容詞や動詞に対して反意語の適用を行うことで、単一のフォースから異なる複数のフォースを導出可能である。例として、図3で示したパターン構造に含まれているフォースの1

つに対し、提案手法を適用した結果を表2に示す。

表2 フォースの変化例

Table 2 Variation Example of force.

変化前のフォース	反意語の適用	強調詞の付加	変化後のフォース
	-	○	開発期間が非常に短い: フォース
開発期間が短い: フォース	○	-	開発期間が長い: フォース
	○	○	開発期間が非常に長い: フォース

表2は、実行する操作を「○」、実行しない操作を「-」で表現し、操作の行否によりフォースがどのように変化するかを表形式で示したものである。以上のように、2通りの操作の組み合わせにより、単一のフォースから異なる複数のフォースを導出することができる。この操作をパターンに含まれる全てのフォースに対して実行することで、多様なバリエーションのパターン構造を導出することが可能となる。

図3で示したまいまい式の構造を例に挙げる。図中の3つのフォースのうち2つに動詞が、残りの1つに形容詞が使用されている。したがって、2つのフォースに対して反意語の適用を、1つのフォースに対して反意語の適用と強調詞の付加の両方を実行可能である。よって、構造変形により得られるパターン構造のバリエーションは、それぞれのフォースに対して適用可能な操作全ての組み合わせから、元のパターン構造を除いた2⁴-1通りとなる。このように、単一のパターン構造に対して構造の変形を行うことで、非常に多くのパターン構造を導出可能であることがわかる。これらのパターン構造群を精査した結果、妥当なパターンとして判断されて導出されたものと、まいまい式との構造の対比を図4に示す。導出された“ハイブリッド走行”という新たなパターンは、まいまい式の構造における“速度を重視しない”というフォースと、“開発期間が短い”というフォースに対してそれぞれ反意語の適用を行うことで得られたものである。それぞれのパターン構造に共通するフォースやコンテキストが確認できることから、提案手法により既知のパターンと関連を持つパターンの抽出が可能であると言える。

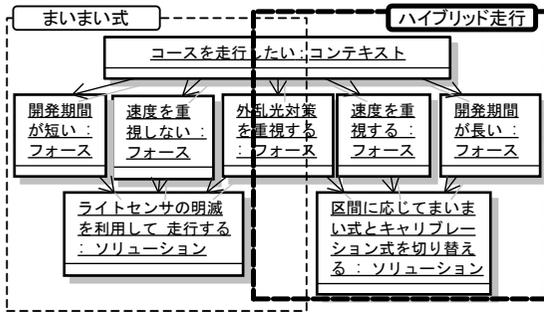


図4 パターン構造の対比

Figure 4 Comparison of Pattern structures

以上の操作をパターンランゲージ内の全てのパターンに対して適用することで、既知のパターンと関連する多様なパターンの導出によるパターンランゲージの拡充が期待できる。

なお、変形対象パターンが類似フォース群を持つ場合、その一部のみのお操作によってフォース群が内部で相反する新パターン候補を導出してしまう可能性がある。その場合、相反を解消する解決策の検討、もしくは、もとの類似していたフォース群の統合を検討する必要がある。また、多数のフォースが含まれる場合はパターン候補数が膨大となり妥当なパターンの識別が困難となるため、重要性が高いと想定されるフォースに絞って変化操作を実施する方法が考えられる。

4. 評価

提案手法が冒頭で述べた研究課題を満たしているか確認するため、第三者による被験者実験を実施した。2012年度のETロボットコンテストに参加した理工学生6名に対し、ETロボットコンテストのためのパターンランゲージに収録の全てのパターンの文書を与え、提案手法を用いたパターン候補の導出を実施した。

実験の結果、被験者グループは30個のパターン候補を導出し、それらを精査して被験者自身の経験を参考として妥当性があると判断される8個のパターン[6]を識別した。所要時間は、パターン候補の導出に1時間程度、パターン候補の精査に30分程度である。

研究課題に対する考察を以下に述べる。

RQ1: 既知のパターンの構造化と構造の変形を通じて新たなパターン候補を導出できるか?

30個のパターン候補が導出され、対象パターンについてある程度の知識を有していれば、第三者が複数のパターン候補を導出できることをETロボットコンテストのためのパターンランゲージについて確認した。

RQ2: 導出パターン候補群において既知のパターンと

関連し妥当と判断されるパターンが含まれるか?

30個のパターン候補から8個のパターンが、経験を参考として妥当と被験者グループにより判断され、それらは既知のパターンへのフォースの変更を通じて導出したものであるためすべて既知のパターンと関連したものであった。得られたパターンの実際の利用における妥当性や有効性の検証は今後の課題である。

RQ3: パターンの構造化と構造の変形を通じたパターンの導出を現実的な時間で達成できるか?

対象パターンランゲージと背景についてある程度の知識を有していれば、提案手法を用いて第三者が複数のパターンを1時間半程度という現実的な時間で導出できることを確認した。より複雑あるいは大規模なパターンランゲージへの適用検証は今後の課題である。

5. 関連研究

Mussbacher[7]らは、ゴール指向要求言語(GRL)を用いてアーキテクチャパターンを形式的にモデル化する手法を提案しているが、構造の変形に伴うパターンの導出には言及していない。対して我々の手法ではその手法を参考に、変形させやすいモデルに基づいてパターンに含まれる項目を明確化すると共に、構造変形により解決策の異なるパターン候補を導出できる。

6. おわりに

我々は、既知のパターンを構造化した上で構造を変形させることで、関連を伴う多くのパターン候補を導出する手法を提案した。提案手法を用いることで、既知のパターンに関連する周囲のパターンを従来手法よりも効率良く導出することに成功した。今後の展望として、提案手法の様々なパターンランゲージへの適用可能性を検証したい。

文 献

- [1] C. Alexander, Sara Ishikawa, Murray Silverstein et al., "A Pattern Language", Oxford University Press, New York, 1977.
- [2] L. Hagge and K. Lappe, "Sharing Requirements Engineering Experience Using Patterns", IEEE Software, Vol.22, No.1, 2005.
- [3] 角谷将司ほか, "ETロボットコンテストを題材としたプロセスが不明瞭な開発におけるパターンマイニングの提案", IPSJ 第177回ソフトウェア工学研究発表会, July, 2012.
- [4] I. Araujo and M. Weiss "Linking Patterns and Non-Functional Requirement", PLoP, 2002.
- [5] K.M. Hansen, et al., "HAZOP Analysis of UML-Based Software Architecture Descriptions of Safety-Critical Systems", NWUML, 2004.
- [5] ETロボコンのためのパターンランゲージ: 追加 http://www.washi.cs.waseda.ac.jp/etrobocon_pattern/new.html
- [7] G. Mussbacher, et al., "Formalizing Architectural Patterns with the Goal-oriented Requirement Language", PLoP, 2006.

Abstract

Constructing a Pattern Language from many patterns facilitates communication and intelligence sharing. However, existing methods of Pattern Mining are so impromptu that it is difficult to mine related patterns. We propose a method that derives related patterns by structuring and utilizing existing patterns in realistic time.

Key words

Software Pattern, Pattern Language, HAZOP