

XMLによるシナリオパターン

徳田 英修[†] 三浦 孝夫[†] 塩谷 勇^{††}

[†] 法政大学 工学研究科 電気工学専攻 〒184-8584 東京都小金井市梶野町 3-7-2

^{††} 産能大学 経営情報学部 〒259-1197 神奈川県伊勢原市上粕屋 1573

E-mail: [†]{i01r3230,miurat}@k.hosei.ac.jp, ^{††}shioya@mi.sanno.ac.jp

あらまし ソフトウェア開発の上流過程における要求分析は、その後の開発の方向を決定付けるという意味で重要な位置づけにある。我々は要求分析時の出力であるシナリオをパターン化し、再利用性を向上させる方法を提案する。我々はシナリオをXMLで記述しタグ変数の概念を導入し検索能力を向上させる。またタグ変数によりシナリオの構造だけでなく意味の類似性からも検索することが可能になる。本稿では例を用いてこの方法を提案する。

キーワード 要求分析, シナリオ, XML

Scenario Pattern using XML

Hidenobu TOKUDA[†], Takao MIURA[†], and Isamu SHIOYA^{††}

[†] Dept.of Elect.& Elect. Engr., HOSEI University 3-7-2, KajinoCho, Koganei, Tokyo, 184-8584 Japan

^{††} Department of Management and Information Science, SANNO University 1573, Kamikasuya, Isehara city, Kanagawa 259-1197 Japan

E-mail: [†]{i01r3230,miurat}@k.hosei.ac.jp, ^{††}shioya@mi.sanno.ac.jp

Abstract The requirement phase in software development is important.If it goes wrong, development will also go wrong to.We patternize the scenario which is an output of the requirement phase, and propose how to improve reusability. We use XML to write scenario.XML will improve reusability of scenario.We use variables over tag.Variables over tag make possible to search a scenario from the similarity of meaning. We propose this method using examples.

Key words Requirement, Scenario, XML

1. 前書き

ソフトウェア上流過程における要求分析は、要求の確定を目的としている。要求を確定するために現在様々な手法がとられており、その中にはシナリオ、UML図を用いた手法がある。UML図は要求を視覚的に見せ、シナリオはユースケースの具体的な流れを示している。これら要求分析の出力を再利用することにより、要求分析にかかるコストを軽減させることが可能である。

我々は既にシステムの静的な構造を示したクラス図を一般化し、そこからパターンを獲得する方法を提案している。[4],[5]

本研究では、要求分析の出力の中のシナリオに着目し、その再利用性を向上させる方法を論じる。シナリオはユースケースの実例であり、自然言語で書かれている。そのためシナリオの再利用を考えた場合その類似性を自動的に検索することが不可能である。再利用をするためにはシナリオを人間が読む必要がありコストが高い。しかしXMLで記述することによりシナリオに構造を持たせた場合は、タグを利用した類似シナリオの検

索が可能となり再利用性が向上するといえる。

本論文の流れとして第2章では我々が扱うシナリオパターンについて定義し、その必要性を説く。続く第3章でシナリオパターンの獲得方法について述べる。第4章でシナリオパターンを利用するために必要な機能について述べ、第5章でそれを実現するためにXML問い合わせ言語の拡張方法を論じる。最後に第6章で結びとする。

2. シナリオの再利用

2.1 シナリオパターンの定義

デザインパターンとはソフトウェア開発の設計における経験をまとめたもので、ソフトウェア設計の際に何度も起きる問題に対する解法を示している。デザインパターンにより、既に成功した設計を一から作る必要がなくなり、他者が様々な状況における問題解決に再利用することによって設計フェーズのコストが軽減される。

本研究で扱うシナリオパターンはデザインパターンより上流過程で用いられる要求分析で出力されるシナリオを再利用を目

的にパターン化したものと定義する。

シナリオとはユースケースの具体的な流れを実例として記述したものである。自然言語で書かれているため再利用を考えた場合、必要となるシナリオを探し出すためには内容を全て人間の手で読まなければならない。シナリオの再利用性を向上させるためには、シナリオを抽象化、一般化する必要がある。具体的すぎると再利用の際、再構築のコストがかかってしまうためである。しかし自然言語で書かれたシナリオを抽象化するとシナリオとしての意味を失ってしまう。この問題はシナリオをXMLで記述することにより解決することが可能である。タグに意味を持たせる事により構造的な観点から一般化することができ、必要なシナリオの検索、類似するパターンの検索に利用することが可能となる。

2.2 シナリオの検索

シナリオパターンを再利用するためには当然欲しいシナリオを検索する必要がある。検索の方法としてはキーワード検索、条件検索、類似検索が挙げられる。XMLで書かれたシナリオならば「<学校>の属性IDが"10001"のもの」などの条件検索がXquery等によって行うことができる。しかし欲しいシナリオを検索する際、「生徒とか学生のようなものを含む文書」や「大学、学部、学科」と似た構造を持つシナリオ（例えば「会社、部署、課」といった類似シナリオを検索する必要も出てくる。

2.3 XMLによるシナリオ

シナリオをXMLで記述することによりシナリオは構造を持つことになる。DTDはXML文書の構造を設計したもので、これを参照すればシナリオの構造を知ることができる。またXMLはタグを自由に作成することができるため、タグの意味による分類が可能である。これは自然言語を意味によって分類するのと比較して効率的である。

XMLはそれだけでタグ同士の親子関係を構造として示しているが、さらにタグ同士の意味関係を関連付けることによってタグの構造を単なる構造ではなく、意味構造として認識できる。その結果、シナリオの意味的な構造から見た類似シナリオの検索が可能になる。例えば<大学><学部></学部></大学>と<大学><大学名></大学名></大学>は構造から見ると同じものである。<大学>の内側にそれぞれ<学部>、<大学名>が内包されている。しかし<大学><学部>と<大学><大学名>の意味的な関係は同じとは限らない。<学部>は<大学>と部分関係にあるが、<大学名>は<大学>と類似関係にあると考える場合、構造は同じでも意味は違うものとなる。

3. シナリオパターン

3.1 類似シナリオ

シナリオをパターン化するためには類似シナリオを考える必要がある。本研究で扱うシナリオはXML化してあり、タグを利用して類似性を判断する。

シナリオの類似には2種類ある。一つはタグの構造の類似であり（図1）、これはDTDによって判断できる。似た構造を持

つということは似たテンプレートで書かれたシナリオであり、同じシナリオパターンの可能性がある。

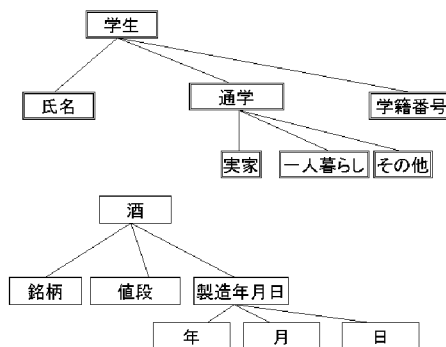


図1 構造の類似

もう一つの類似はタグの意味による類似である。例えば図2では<学生>と<生徒>は似た意味ととることができる可能性がある。

構造の類似だけでは類似パターンと判断するには不十分であり、最終的には意味の類似の確認をする必要がある。

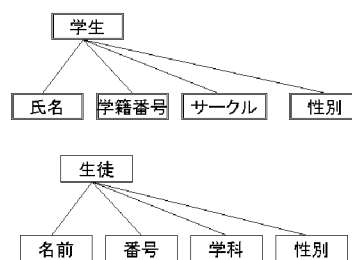


図2 意味の類似

しかし意味の類似は主観的な判断が必要であり完全な自動化は不可能である。意味の類似を可能な限り自動化するためにはその判断基準が必要となる。

3.2 タグによる類似性の判断

本研究では、タグ同士の意味関係の記述により意味的な類似性を判断する。タグの関係は類似関係、親子関係、等価関係、部分関係の4つとし、シナリオとは別に記述する。意味関係の記述によりタグ同士の関係からの類似性判断が可能となり、パターン検索の幅が広がる。例えば図3の場合、二つのシナリオは構造は同じだがタグ同士の意味関係が異なるため類似シナリオパターンではない。図4では二つのシナリオは構造、意味ともに一致し同じシナリオパターンであると考えられる。

タグを意味関係でまとめると意味関係が重複したものがでてくる。例えば図5ではタグ同士の類似関係、親子関係が示されている。a,c,dとe,bはそれぞれ類似関係にある。これらはそれぞれ意味構造として同じであることを示しているので、図6でa,c,dはAに、e,bはBに一つの変数としてまとめることが可能である。これをタグ変数とする。また図5でc,dは共にbとの親子関係を持っている。よって図6のようにここでもc,dは

- $\langle A \rangle \Leftrightarrow \langle B \rangle \dots$ 親子関係
- $\langle A \rangle \Leftrightarrow \langle C \rangle \dots$ 類似関係

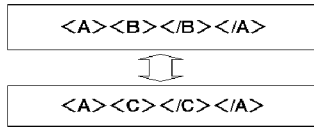


図3 構造のみ類似

- $\langle A \rangle \Leftrightarrow \langle B \rangle \dots$ 類似関係
- $\langle A \rangle \Leftrightarrow \langle C \rangle \dots$ 部分関係
- $\langle B \rangle \Leftrightarrow \langle D \rangle \dots$ 部分関係

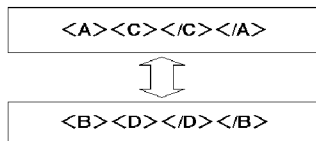


図4 意味と構造が共に類似

タグ変数 A と変換している。タグ変数によって抽象化されたシナリオがシナリオパターンとなる。

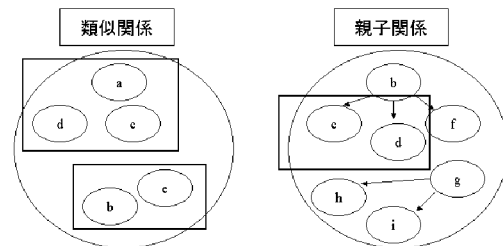


図5 意味関係の格納

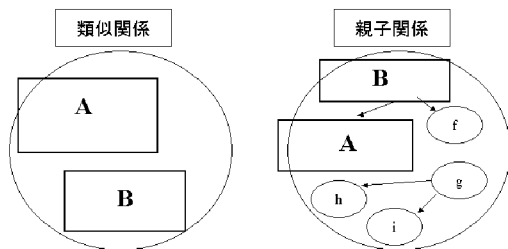


図6 タグ変数

4. シナリオパターンの利用

4.1 類似シナリオ検索

XMLで書かれたシナリオは構造を持つことになり、DTDを参照すればシナリオの構造が判断できる。

類似シナリオの検索をする場合、システムはまず関係記述に含まれるタグを探し、その関係を調べる。この際参照するのはタグ変数である。続いて関係記述に含まれるタグ同士の構造的

関係をDTDを参照して調べる。関係記述が同じタグであったとしても、タグ同士の構造的関係が親タグ子タグ関係の場合と同じ親を持つ子タグ同士の関係の場合では意味合いが違ってくる。構造的関係、意味的関係の両方から判断し、両者が共に同一のものが類似パターンであると言える。このようにシナリオパターンの検索は、タグの構造とタグ同士の意味関係の検索によって行われる必要がある。

4.2 XML問い合わせ言語

XML文書形式で格納された情報を問い合わせるために様々なXML問い合わせ言語が存在する。現在はその中でも特にXQueryが標準候補として注目されている。XML問い合わせ言語を利用することにより大量のXML文書の中から必要な情報を抽出することが可能になる。

これらのXML問い合わせ言語を利用するには文書のタグ構造を知っていることが前提であり、タグ自体の検索に使用することはできない。

本研究で用いるXMLで書かれたシナリオパターンの検索を考えた場合、従来の検索だけでなくタグ自体の検索が必要である。例えば子タグが親タグの部分集合になっているシナリオパターンを探すにはタグの検索が必要となってくる。そのためにはXML問い合わせ言語の拡張が必要となる。

5. Xquery 拡張

```
<schools>
  <school ID="10001">
    <name>法政大学</name>
    <abbreviation>法大</abbreviation>
    <departments>
      <department>
        <name>工学部</name>
        <courses>
          <course>
            <name>情報電気電子工学科</name>
            <name>建築学科</name>
            .....
          </course>
        </courses>
      </department>
      <department>
        <name>法学部</name>
        .....
      </department>
    </departments>
  </school>
</schools>
```

図7 シナリオ sample.xml

図7はXMLで書かれたシナリオである。図8はシナリオ上部のタグから意味関係を抽出し、タグ変数を与えてシナリオパターンとしたものである。ここでユーザは<scool>、<name>、<abbreviation>が類似関係、<scool>と<departments>が部分関係にあると判断している。<name>と<abbreviation>は類似関係にあるので、タグ変数Bが割り当てられている。ここで<scool>だけタグ変数Aとなっているのは<name>、<abbreviation>と違い、<departments>と部分関係をもっているためである。

"C part A" "B similar A"はタグの意味関係を記述したものである。"C part A"はCはAと部分関係にあることを示し、"B similar A"はBとAが類似関係にあることを示している。

DTDで示された構造と、タグ変数が示しているタグ同士の意味関係からシナリオパターンの検索をXML問い合わせ言語

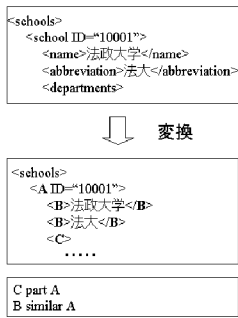


図 8 シナリオパターン

によって行いたい。Xquery では図 9 のようにタグのパスを指定することによって問い合わせを実行する。そのためタグ自体の検索を行うためには拡張が必要となってくる。

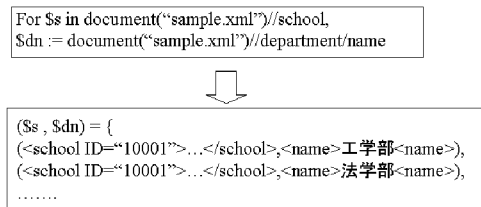


図 9 Xquery による検索

我々は問い合わせるタグを \$\$ で示すことにする。また従来パス表現式を記入していた部分にタグの関係を表す式を記入することができれば、タグの検索が可能となる。ここでいうタグの関係を表す式というのは similar, equal, part, parent でありそれぞれ類似関係、等価関係、部分関係、親子関係を表している。

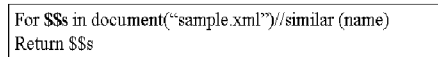


図 10 Xquery の拡張検索 1

例えば図 10 では <name> と類似関係にあるタグを検索している。<name> はタグ変数 B なので、同じタグ変数 B が割り振られているタグと類似関係が式で示されているタグ変数 A が検索される。具体的には <scool>、<abbreviation> である。図 7 に示されているさらに下の階層の <department> や <course> も <name> と類似関係ならば検索されることになる。

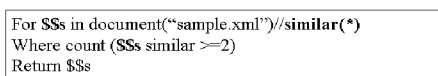


図 11 Xquery の拡張検索 2

図 11 では類似関係が 2 つ以上あるものを検索している。この場合、<scool>、<name>、<abbreviation> が類似関係となっており 2 つ以上の条件を満たしているため検索される。

以上のように Xquery を拡張すれば従来の検索に加え、シナリオパターンの検索が可能となる。従来パス表現式を記述していた部分にタグの意味関係を示す関係式をいれ、問い合わせるタグは \$\$ で示す。関係式で示された条件を元にシステムはシナリオパターンを検索し、必要な条件を満たすシナリオパターンを検索する。

6. 結 び

本研究では要求分析における出力の一つ、シナリオのパターン化について論じた。シナリオはユースケースの具体例を自然言語で示したものであるので抽象度が低く、再利用に向かない。そこで我々はシナリオを XML で記述することによりシナリオに構造を持たせる方法を提案した。さらに XML で記述されたシナリオに含まれるタグ同士の関係を記述しシナリオに意味構造を持たせた。タグ同士の関係が同じものはタグ変数としてまとめシナリオの抽象度をあげシナリオパターンを抽出した。ここで抽出したシナリオパターンを XML 問い合わせ言語を利用して検索するために、Xquery の拡張について論じた。従来パス表現式を記述していたところにタグの関係を表す式を記述し、タグの意味関係からの検索をした。ユーザがシナリオを XML で記述しタグ変数を与えることによって、シナリオのタグによる検索が可能となりシナリオをシナリオパターンとして再利用することが可能となる。

謝 辞

本研究の一部は文部科学省科学研究費補助金 (課題番号 14580392) の支援による。

文 献

- [1] Serge Demeyer, Stephane Ducasse and Sander Tichelaar: "A Pattern Language for Reverse Engineering", proc. 4th European Conference on Pattern Languages of Programming and Computing, 1999.
- [2] Erich Gamma, オブジェクト指向における再利用のためのデザインパターン, ソフトバンクパブリッシング, 1999.
- [3] Stephen R. Schach, Object-Oriented and Classical Software Engineering, McGraw Hill College Div, 2001.
- [4] 徳田 英修, 白岩 政太郎, 三浦 孝夫: "要求仕様のパターン獲得", DBWS 2002.
- [5] Tokuda, H., Shirowa, M., and Miura, T.: On Pattern Acquisition of Domain Knowledge, *Reuse in Object-Oriented Information Systems Design (OOIS)*, 2002