

1Q-2

大規模 XML 文書のデータバインディングにおける マッピングファイルの作成支援ツールの開発

衣川 勇樹[†]

内藤 広志[‡]

大阪工業大学 情報科学研究科[†]

大阪工業大学 情報科学部 情報メディア学科[‡]

1 はじめに

XML データバインディングは、XML 文書と Java オブジェクトをコードを書くことなく対応づける技術である。これには 1) XML スキーマに対応した Java クラスを生成する手法と 2) XML 文書をユーザ定義の Java オブジェクトに直接マッピングする手法があるが、大規模な XML 文書においては 1) と 2) とともに実用性が乏しい。そこで、2) の手法において設定ファイルを作成することが困難であるという問題を解決するためのツールを開発した。

本発表では、大規模スキーマを用いた場合の 1) と 2) の問題点について指摘し、作成したツールによる解決策について述べる。

2 XML データバインディングの手法

2.1 コード生成

コード生成は、XML スキーマに対応する Java クラスを自動生成し、そのクラスを用いて XML 文書と Java オブジェクトの相互変換を行う手法である(図 1)。ツールによって、XML スキーマから Java クラスが生成されるので、Java クラス作成の手間が要らず、すぐにアプリケーションの開発に取りかかることができるという利点がある。

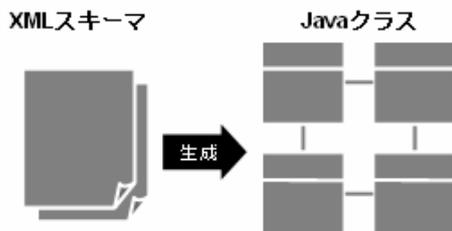


図 1 コード生成

2.2 マッピング

マッピングは、設定ファイルを用いて、XML 文書を直接ユーザ定義の Java クラスにマッピングする手法である(図 2)。Java クラスをユーザが定義できるため柔軟性が高く、スキーマを変更してもアプリケーションに影響が出にくい。

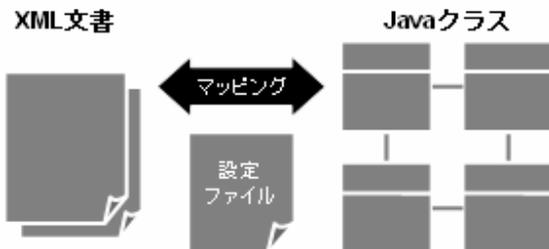


図 2 マッピング

3 バインディングの問題点

B2B には XML 文書が広く用いられるが、中でも UBL [1]のような文書標準は、非常に規模の大きなものである。このような大規模な XML 文書を用いた場合、バインディングに次に述べるような問題が生じる。

3.1 コード生成の問題点

コード生成は、XML スキーマの全ての型と要素に対して Java クラスを生成する。したがって、大規模な XML スキーマを入力した場合、生成される Java クラスが膨大な数となる。例えば、Sun 社の XML バインディングツールである JAXB[2]に UBL の Order のスキーマを入力した場合、200 以上のクラスが生成される。また、アプリケーション開発時、XML 文書のデータへアクセスするために、Java オブジェクトに対して DOM 同様の階層的なアクセスをする必要があり、XML データバインディングの目的である開発の容易性が大きく損なわれる。

3.2 マッピングの問題点

マッピングでは、ユーザが Java クラスを定義するので、コード生成のような問題はないが、大規模な XML 文書を用いるため、XML 文書全体の把握ができないと、設定ファイルの作成が非常に困難となる。また、ユーザはアプリケーション実行時まで設定ファイルが正しいかどうかかわからず、正しいマッピングを設定するのに、多くの時間が必要となる。マッピングによる XML データバインディングを行える Castor[3]の場合、O2XMapper[4]という設定ファイル作成支援ツールがあるが、Java クラスや XML 文書の要素/属性を入力フォームに直接入力する必要があるため、やはりこれらの問題を解決することはできない。

Development of supporting tool for Databinding with large-scale XML document

[†] Yuki Kinugawa, Graduate School of Information Science and Technology, Osaka Institute of Technology

[‡] Hiroshi Naito, Faculty of Information Science and Technology, Osaka Institute of Technology

4 設定ファイル作成支援ツールの開発

3.2 で述べた XML データバインディングにおけるマッピング手法の問題点を解決するため、マッピングファイルの作成支援ツールを統合開発環境 Eclipse[5]のプラグインとして開発した。

4.1 Castor を用いたマッピング

本研究では XML バインディングツールとして、3.2 で述べた Castor を使用した。マッピングの設定ファイルは、Java クラスのフィールドと XML 文書の要素/属性とを関連付けている (図 3)。

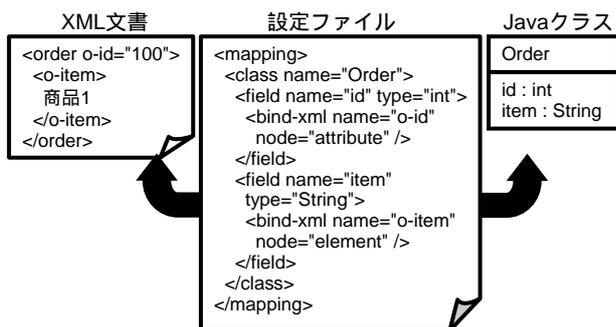


図 3 設定ファイルによるマッピング

4.2 開発したツールの機能

本研究で開発した設定ファイル作成支援ツールは 2 つの機能を提供する。

1) 複数ビューによる対象の表示・選択機能

3 つのビューで設定対象を選択して、マッピングの設定を行う。以下に手順を示す。

- パッケージ・エクスプローラー (図 4 の) で Java クラスファイルを選択する。
- フィールドビュー (図 4 の) で設定したい Java のクラスとフィールドを選択する。ここで、マッピングエディター (図 4 の) で表示されたファイルに class 要素と field 要素が追加される。
- パッケージ・エクスプローラーで XML スキーマファイルを選択する。
- 先に選択したフィールドと関連付けたい要素/属性を要素ビュー (図 4 の) から選択する。ここで、マッピングエディターで表示されたファイルに bind-xml 要素が追加される。

このように、設定ファイルの全ての値はビューで選択して指定することができ、誤った値が入力されることがない。

また、フィールドや要素の詳細な情報はプロパティビュー (図 4 の) で参照することができる。

2) XML スキーマからの要素/属性ツリー生成機能

機能 1)において、手順 d)で用いた要素ビューは、手順 c)で選択した XML スキーマから生成している。

これは、XML 文書の要素/属性を、出現回数やデータ型の情報などを含め、ツリー構造で表示するものである。この要素ビューにより、ユーザは大規模なスキーマからも文書の全体像を容易に把握することができる。

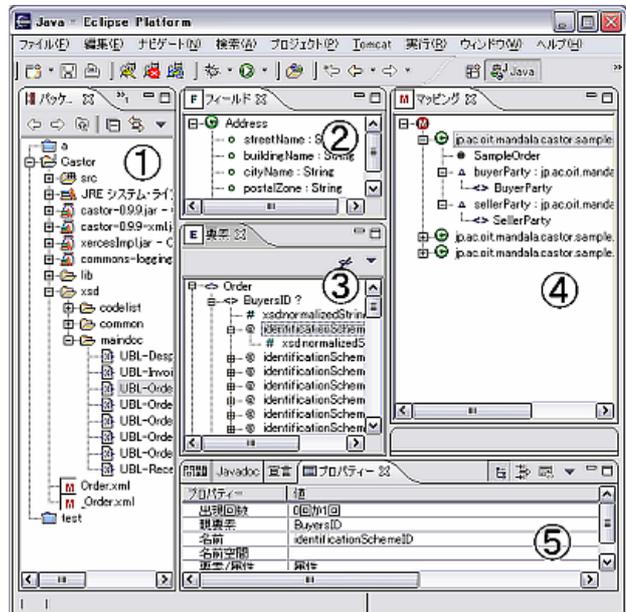


図 4 設定ファイル作成支援ツール

5 まとめ

本研究において開発したツールを用いると、ビューから関連付けたい Java フィールドや XML 要素を選択していくだけで、マッピングの設定ファイルを作成することができる。XML 文書の全体を把握することができ、大規模な XML 文書でも設定ファイルの作成が容易になる。

しかし、マッピングによる XML データバインディングでは、XML スキーマで定義された全ての要素/属性について関連付けを行う必要はないため、実際に XML 文書を Java オブジェクトに変換した際、未設定の情報が落ちる可能性がある。そのため、XML 文書を入力したときに、未設定の要素/属性の情報落ちをチェックする機能を追加する必要がある。また、XML データバインディングを用いて、実際に大規模な XML 文書に対応したアプリケーションを作成し、本ツールの有用性を検証する必要がある。

参考文献

- [1] OASIS Universal Business Language (UBL) TC
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl
- [2] Java Architecture for XML Binding (JAXB)
<http://java.sun.com/webservices/jaxb/>
- [3] The Castor Project
<http://www.castor.org/>
- [4] O2XMapper
<http://o2xmapper.sourceforge.net/>
- [5] Eclipse.org home
<http://www.eclipse.org/>