JAIST Repository
https://dspace.jaist.ac.jp/

Title	XMLで表現されたCプログラムの静的解析ツールの設計と実現
Author(s)	川島, 勇人
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1532
Rights	
Description	Supervisor:権藤 克彦, 情報科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

XMLで表現されたCプログラムの静的解析ツールの設計と実現

川島 勇人 (010035)

北陸先端科学技術大学院大学 情報科学研究科

2002年2月15日

キーワード: XML, CASE ツールプラットフォーム, データ統合, DTD 設計.

1 背景

ソフトウェア開発、保守において、CASE(Computer Aided Software Engineering) ツールが必要不可欠である。しかし、CASE ツール開発のコストは非常に高い。それは、たいていの場合、その CASE ツールの内部データを他の CASE ツールで使えないことにある。それゆえ、効率的に CASE ツール間でのデータ共有やデータ変換を可能にする技術、つまり、データ統合技術を発見、開発することが大きな課題となっている。

2 目的

この問題を解決するために、CASE ツールプラットフォームが提案、構築されてきた (例えば、Sapid)。CASE ツールプラットフォームとは、CASE ツールに共通なデータや機能を提供する基盤となるソフトウェアシステムである。本研究では、XML(Extensible Markup Language) の CASE ツールプラットフォームへの適用が、有効であることを示す。適切に設計した DTD(Document Type Definition) と、DOM(Document Object Model) や XSLT(Extensible Stylesheet Language Transformations) を始めとする XML 関連技術の適用が、CASE ツールの開発コストを大幅に削減すると期待できる。

我々の目標は、CASE ツールのための柔軟で使い易いデータ変換フォーマットの構築である。これは、CASE ツール開発を容易にする重要な要素である。特に、我々が注目するのは、XML による下流 CASE の統合である。なぜなら、ソフトウェア開発コストは、その大部分がテストや保守に費やされるからである。また、XML の下流 CASE への応用例は、まだほとんどない。

Copyright © 2002 by Hayato Kawashima

3 CASE ツール開発における XML の利点

CASE ツールのための共通フォーマットのアイディアは、新しいものではない。例えば、CDIF(CASE Data Interchange Format) や PCTE(Portable Common Tool Environments) が、すでに提案されている。これらの技術は、データ統合に貢献しているが、未だ CASE ツール開発に広く使われてない。これは、本質的にデータ統合が難しいからである。

我々の目標を達成するために、主に次に挙げるの点から、XML は有効である。

- プログラム構文構造は、XML のエレメントの入れ子構造として自然に表現でき、プログラム要素間の関係は ID/IDREF リンクとして表現できる。
- 妥当な XML 文書として扱われないが、整形式な XML 文書として表現すれば、不完全なデータ (例えば、バグのあるコード) でさえも扱うことができる。
- XMI(UML 図の XML 表現形式) の標準化が進んでいる。下流 CASE ツールに XML を適用すれば、上流と下流の間のデータ統合が期待できる。

4 実現した CASE ツールプラットフォームの概要

本研究では、我々の目標を達成する第一歩として、ANSI C プログラムのみをターゲットにした CASE ツールプラットフォームを実現した。XML が実用的に活用できるか確認するために、フルセットの ANSI C をサポートすべきと考えている。実現した CASE ツールプラットフォームは、(1)ANSI C 構文規則を表現した ACML、(2) ソースプログラムを ACML 文書に変換するトランスレータ XCI、(3)XML 技術を基に開発した CASE ツール、から成る。

5 CASE ツール作成実験

XML を用いた CASE ツールプラットフォームを基に、プログラムスライシングツールとクロスリファレンサを実験的に作成した。これらの実験は、CASE ツールプラットフォームの有効性を計る例題として適切である。その主な理由は、これら 2 つの CASE ツールが、次に示す特徴を持つからである。

- 他の CASE ツールに比べ、非常に基本的、かつ必要性の高い静的解析ツールである。
- 解析において、プログラム要素間の関係を処理する必要がある。

これらの実験において、各々の作成は開発者 1 人で、わずか約 2 週間ですんだ。これに対し、XCI の ANSI C パーサと静的意味解析器の実現には、開発者 1 人で、2 カ月を費やした。それゆえ、ACML と XCI を使ったことで、本来、必要であるはずの 2 カ月の開発期間を削減できたことになる。

6　議論

我々は、上で述べた実験を通じて有益な経験をした。第一に、ACML には改良の余地があることを確認した。例えば、あるノードに対して、自分が何番目の兄弟であるかを示す属性がない。これは、木構造内の走査、特に上方へ辿るときに便利である。

第二に、DOM や XSLT の利便性は高かったが、CASE ツール開発において、共通に使用できるライブラリを構築する必要性を確認した。例として、次の操作を挙げる。

- 特定の識別子を持つステートメントをすべて抽出する。
- 特定の識別子を含んだ最も深い部分のステートメントを抽出する。

第三に、DTD の利便性を再認識した。それは、ACML がデータ交換の簡潔な仕様として機能したので、実際に XCI 開発者と CASE ツール開発者のやり取りは、ACML を定義した DTD で、ほとんどすんだ。

7　まとめ と 今後の課題

我々は、XML を用いた CASE ツールプラットフォームを提案した。まず、ACML を定義した。これは、抽象構文木と静的意味 (例えば、型やシンボル) を表現することができる。それから、CASE ツール作成実験として、この CASE ツールプラットフォームを基にプログラムスライシングツールとクロスリファレンサを各々開発者 1 人、開発期間約 2 週間で実現した。XCI の ANSI C パーサと静的意味解析器を実現した約 2 カ月を考慮すると、多少粗い議論にはなるが、開発期間を 2 カ月間、削減した。よって、この種の小さな CASE ツール開発では、XML を用いる有効性が確認できた。

今後の主な課題を以下に示す。

- 動的解析に対応する ACML の拡張
- Java や C++など他の言語へ応用
- ACML を使った他の CASE ツールの作成 (例えば、テストケース生成器)
- 前処理命令や字句情報 (例えば、コメント、コーディングスタイル) の保存、システムコール、インラインアセンブラへの対応
- ACML と XMI の連携手法の構築
- 未完成 (バグだらけ) のプログラムやプログラマの意図に対する適切な表現の定義