

## ドメイン特化型言語を用いた分野横断知識の 連携手法の研究

佐々木, 晃 / SASAKI, Akira

---

(雑誌名 / Journal or Publication Title)

科学研究費補助金研究成果報告書

(開始ページ / Start Page)

1

(終了ページ / End Page)

4

(発行年 / Year)

2011-05

## 様式 C-19

## 科学研究費補助金研究成果報告書

平成 23年 5月 30日現在

機関番号：32675

研究種目：若手研究（B）

研究期間：2009～2010

課題番号：21700041

研究課題名（和文）ドメイン特化型言語を用いた分野横断知識の連携手法の研究

研究課題名（英文）A Study for Supporting Integration of Knowledge across Interdisciplinary Fields using Domain Specific Languages

研究代表者

佐々木 晃 (SASAKI AKIRA)

法政大学・情報科学部・准教授

研究者番号：90396870

研究成果の概要（和文）：高度な問題解決を行うために、異分野間にまたがる情報や知識を統合して扱うことが可能なソフトウェアアプリケーションが求められている。しかし、それぞれの分野における言語の違いや、情報の扱い方の差異を吸収しながら、そのようなアプリケーションを開発することは簡単ではない。本課題では、様々な分野の知識処理を可能とする「ドメイン特化型言語」を利用し、異分野知識を連携させるソフトウェアを効果的に作成する手法について研究を行った。

研究成果の概要（英文）： Software application based on knowledge or information across multiple fields is difficult to develop since it requires to absorb the differences of those languages and methods to treat information in each field. In this study, a methodology is investigated that enables to effectively develop software systems using multidisciplinary knowledge through implementing domain specific languages.

交付決定額

（金額単位：円）

|        | 直接経費      | 間接経費    | 合計        |
|--------|-----------|---------|-----------|
| 2009年度 | 1,300,000 | 390,000 | 1,690,000 |
| 2010年度 | 1,200,000 | 360,000 | 1,560,000 |
|        |           |         |           |
|        |           |         |           |
| 総計     | 2,500,000 | 750,000 | 3,250,000 |

研究分野：情報科学

科研費の分科・細目：情報学・ソフトウェア

キーワード：ドメイン特化型言語、属性文法、言語開発環境、プログラミング言語、コンパイラ

## 1. 研究開始当初の背景

問題解決や意思決定はある特定の分野（ドメイン）だけの知識にフォーカスして行おうとすると、局所解にしか到達しない。逆に、様々なドメインにおける知識を共有して連携させることは、新しい価値を持った知識や情報を生み出す。本課題の全体的な目標は、複数の分野の知識を横断的に扱うための系統的手法とプラットフォームの確立である。

情報化社会における昨今、異分野の知識を統合するという社会的ニーズは大きいですが、分野横断的な知識を扱うことは簡単ではない。

知識の統合に対する社会的ニーズに対しては、World Wide Web上の知識やリソースを連携させることで新たな価値を利用者に提供するWebサービス技術などが生まれた。現在、ネットワーク上の多種多様な知識を組

み合わせたシステムを構築するためには、データベースアクセスや XML (データの標準化法) との親和性の高い JavaScript 言語のようなプログラム言語を利用して行う手法が代表的である。

しかしこれらは、各種の「生の」データを汎用プログラム言語によってアドホックに処理するものであり、可読性や可搬性が低いという点で、このように構築されたシステムは脆弱であると言わざるを得ない。また、なによりも汎用プログラム言語を書くことができないユーザは、知識を連携した問題解決を計算機上で行うことができない。

## 2. 研究の目的

背景の項で述べた問題を解決するために本課題では、知識連携を可能とするアプリケーションをドメイン特化型言語 (Domain Specific Languages, DSL) を中心としたフレームワーク上で構築する手法を提案する。ドメイン特化型言語は、特定の領域の問題を、その領域に適したモデルの記述方法で表現するための言語であり、近年注目されている。一般には、汎用プログラミング言語とは異なるものであるが、ドメイン特化型言語で書かれたモデル記述は、適切な処理系によってコンピュータ上で実行可能である、という特徴をもつ。これにより次の二つの実用上の利点がある：1, 連携する知識および問題解決のレベルに応じた言語を作成することが可能である。2, 問題解決を行う専門家が、たとえ汎用プログラム言語を書くことができない場合でも、ドメイン特化型言語を用いて問題 (すなわちプログラム) を記述し、コンピュータ上で解決 (実行) することが可能となる。

## 3. 研究の方法

本課題では (1) 知識連携アプリケーション作成フレームワークの設計 (2) フレームワークの中心となる知識連携処理向けドメイン特化型言語の定式化、の 2 点について研究を行うことで提案する手法の実現を目指した。様々な知識を連携して使用するためには、次の事項を系統的かつ容易に行うことが必要である：(a) 知識データの中から、問題解決に必要な知識を必要な抽象レベルで取り出せること (b) 取り出した知識を言語要素として利用できるドメイン特化型言語処理系の実現が可能であること、および (c) 利用者のプログラム能力に応じた言語開発環境を実現すること。

(a) (b) では、知識の取り出しや、その知識を扱うドメイン特化型言語処理系の実装を、知識表現仕様および言語仕様から自動生成

して行うことを目指す。通常言語仕様を定義する際には、その言語で使えるデータ型の仕様が含まれるが、本手法では、連携すべき知識表現に基づくデータ型を拡張言語要素として組み込むこと (プラグイン) を想定する。

(c) における、コンパイラ、エディタ等の言語開発環境の開発スキームは、研究代表者の過去の研究成果である、ドメイン特化型言語開発環境フレームワークを用いることで、効果的に研究を進める。最終的には、提案したフレームワークおよび定式化手法によってドメイン特化型言語を実際に実装し、この上でモデルケースとなる知識連携アプリケーションの開発を行う。

## 4. 研究成果

### (1) 概要

本課題では上記の研究方法に基づいて研究を行った。主な研究成果は次の①②③である。

#### ① 知識連携のためのドメイン特化型言語フレームワークの設計:

ドメイン特化型言語のフレームワークの核となる定式化手法について研究を行った。ドメイン特化型言語に関する調査は研究成果②⑤などで言及している。定式化手法は、木文法および属性文法を核としたものであるが、これに関しては研究成果①、③、④、⑥で示した。

#### ② 拡張可能なドメイン特化型言語およびそのエディタの生成システム:

ビジュアルエディタの作成を支援するツールを Java 言語および Swing GUI ライブラリにより実装した。エディタ作成者は、木文法の構文定義を与えることで、その言語向けのエディタの基本部分を構成したことになる。GUI の表現方法は、4 節で述べたようにいくつかのバリエーションがあり、各導出レベルでの表現方法を指定する必要がある。これをカスタマイズすることで、エディタを自動生成することができる。これに関しては研究成果①、③、⑤で示した。

#### ③ 知識連携の基盤シミュレーションシステムの設計と実装:

提案手法を用いて、知識連携の基盤となるシミュレーションシステムのプロトタイプとして、インタプリタおよびエディタの一部の実装を行った。これは、既存の社会科学シミュレーションシステム SOARS に基づいたシステムである。さらに、この基本システムを言語拡張する方法で、形式化された知識表現に基づいてシミュレーション可能な言語処理系の試作を行った。これらの成果の一部は、

研究成果①、③で示した。

## (2) ドメイン向けシミュレーションシステムの開発手法

本項では主に研究成果①および③に基づき、特定分野の知識に基づいたドメイン向けエージェントシミュレーションを効果的に開発する手法の概要について述べる。

汎用的なエージェントシミュレーション記述言語によって記述されたシミュレーションモデルの内容や実行結果を解釈する場合には、汎用言語によるモデル記述を問題固有のより高次の言葉や概念で解釈し、理解し直す必要がある。これは、シミュレーションの最終的なユーザーが、モデルの開発者でない場合、すなわち汎用シミュレーション言語の知識を持たないユーザーがシミュレーションを解釈したい場合に不都合となる。例えば、次のような場合が考えられる。

- (1) シミュレーション結果を広く公表する。
- (2) 異分野の専門家同士がシミュレーションをコミュニケーションツールとして用いる。

このような場合に生ずるシミュレーションの解釈の難しさは、シミュレーションモデルの開発者が汎用的に利用する言語と、最終的なユーザーが理解できる言語との間のギャップが、大きな要因の一つである。このような言語のギャップとして、(1) 言語ボキャブラリの違い (ボキャブラリギャップ) (2) 表現する問題構造のギャップ (構造ギャップ) が存在する。本研究では、このようなギャップを「ドメイン向けに特化したシミュレーション言語」の導入によって埋める手法を提案した。この手法により、特定の分野の知識を用いた複数のシミュレーションを効果的に開発することが可能となる。

ドメイン向けシミュレーション言語の実装は、汎用シミュレーション言語に比べ比較的容易であると考えられるが、実際には次の点に考慮する必要があり難しい問題が含まれる。(1) シミュレーションシステムには、言語処理系を含む開発環境を揃える必要がある。このような環境をすべて揃えるにはコストがかかる。(2) 新しい言語を作成する際には言語仕様の更新が頻繁に起こる可能性があるが、それにともない、開発環境すべての更新が必要となる。

そこで本研究では、ドメイン向けエージェントシミュレーションシステムを効果的に開発するフレームワークを提案する (図 1)。本フレームワークでは、開発者は、以下のア

プローチでドメインに特化したエージェントシミュレーション言語とその環境の開発を行う。(1) エージェントベースシミュレーションの開発環境 (ツール群) をシミュレーション言語の仕様に基づいて開発する。(2) (1) の方法で開発したベースとなる汎用シミュレーションシステムをもとに、その上にドメイン向けの拡張を仕様に施す。

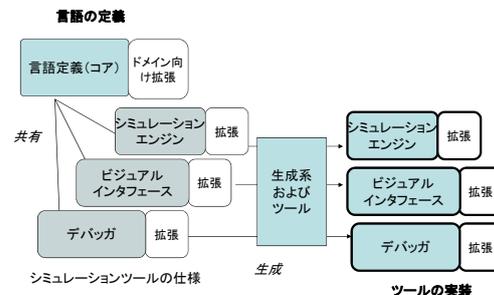


図1 フレームワーク概要

本アプローチでは、仕様記述に基づいた開発を想定している。これにより生成系などのツールの利用が可能であり、効果的にインプリメンテーションが得られるため、それぞれのツール開発におけるコスト軽減が図れる。これにより前述の1番目の問題点が解決される。さらに、このような生成系の利用では、モデル (仕様) と実装の対応がとりやすいため、言語仕様への拡張や変更への対応が効果的に行える利点がある。これは前述の2番目の問題を軽減する。

エージェントシミュレーションシステムでは、シミュレーション言語の処理系 (シミュレーションエンジン) はもとより、そのビジュアルインタフェース (エディタ)、デバッガや、シミュレーションエンジンを取り巻く実行環境 (シェル) など、様々な機能が必要となるが、本アプローチでは、開発者はそれぞれの機能を開発環境の中の独立したツールとして提供する。その際、これらの独立したツールを共通の言語仕様をコアとした統一的な枠組みの中で開発可能であることが、本フレームワークの特色となる。

本研究では、言語の仕様記述からビジュアルインタフェースを自動生成する「インタフェースデザイナー」のプロトタイプシステムを作成した。我々は(1) Java による実装 (2) Squeak Smalltalk による実装の2種類を作成し、本手法の実現可能性を確認した。図2は、JavaのSwingによるビジュアルインタフェース生成システムによって生成された実行結果である。図3は Squeak Smalltalk による

インタフェースデザイナーによる、ビジュアルインタフェースの開発の様子を示したものである。これらの実装では、木文法によるシミュレーション言語の定義から、ビジュアルインタフェースのプロトタイプを自動生成する。ビジュアルインタフェースの開発者は、このプロトタイプを基に、外観をカスタマイズしていく。カスタマイズは、UI 部品を直接的な操作によって、細かいプロパティを設定することによって行う。

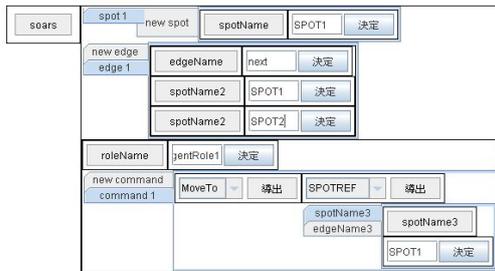


図2 Java Swingによるビジュアルインタフェース実装

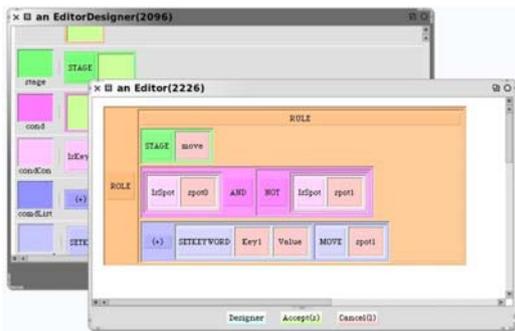


図3 Squeak Smalltalkによるインタフェースデザイナー

図4は、図2、図3のインタフェースを自動生成するための木文法によるシミュレーション言語の定義の例である。

```
spot => SPOT t_spotName
agent => AGENT t_agentName t_initSpotName
        t_roleName
spotRel => SPOTREL edge*
edge => EDGE t_edgeName
        t_spotName t_spotName
role => ROLE t_roleName
        condition* command*
condition => ISSPOT t_spotName
        | IS t_key t_value
command => MoveTo spotRef
        | SETKEY t_key t_value
spotRef => SPOTREF t_spotName t_edgeName
        | CURSPOT t_edgeName
```

図4 シミュレーション言語の仕様

## 5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[学会発表] (計6件)

①市川 寛、佐々木 晃、拡張可能なドメイン特化型言語のエディタの生成手法、情報処理学会全国大会 2011年3月2日、東京工業大学

②篠崎 雄一郎、佐々木 晃、教育向けプログラミング言語を応用した高級言語学習のための環境の開発、情報処理学会全国大会 2011年3月3日、東京工業大学

③佐々木 晃、出口 弘、ドメイン向けエージェントシミュレーションシステムとその実現基盤、合同エージェントワークショップ & シンポジウム 2010、2010年10月28日、新富良野プリンスホテル

④Akira Sasaki, Hiroshi Deguchi An Efficient Development Method for Domain Specific Simulation Systems, The Third World Congress on Social Simulation, 2010年9月6日、ドイツゲント大学

⑤佐々木 晃、市川 寛、田沼 英樹、GUIコンポーネントに基づく視覚的言語に対するエディタの自動生成、情報処理学会プログラミング研究会、2010年3月16日、電気通信大学

⑥佐々木 晃、市川 寛、ドメイン特化型言語のための開発支援手法、電子情報通信学会ソフトウェアサイエンス研究会、2009年12月17日、香川大学

## 6. 研究組織

(1) 研究代表者

佐々木 晃 (SASAKI AKIRA)

法政大学・情報科学部・准教授

研究者番号：90396870