

行動認識アプリケーション向けドメイン特化型言語 LOARA の設計と評価

一野 浩太朗^{†1} 久住 憲嗣^{†2} 井上 創造^{†3}
中西 恒夫^{†4} 福田 晃^{†4}

行動認識アプリケーションの開発上の課題として、1) 開発に必要な知識がプラットフォームとパターン認識の2つの分野をまたいでおり、高度な技術が必要となる、2) 精度を向上させるにはパラメータチューニングを行いながら再実装を繰り返す必要があるため工数がかかる、が挙げられる。そこで行動認識アプリケーション開発にドメイン特化型開発を導入することで、課題の解決をはかる。ドメイン特化型開発では、DSL と呼ぶ特定ドメインのアプリケーションを効率的に開発するための言語を開発しアプリケーション開発を行う。本論文では、行動認識向け DSL である LOARA の設計、および実装して評価を行った結果を示す。

Designing and evaluation of LOARA, a Domain-Specific Language for Activity Recognition Applications

KOTARO ICHINO,^{†1} KENJI HISAZUMI,^{†2} SOZO INOUE,^{†3}
TSUNEO NAKANISHI^{†4} and AKIRA FUKUDA^{†4}

Developmental issues in activity recognition applications is shown below. 1) Knowledge is necessary for developing two field, platform and pattern recognitions. Therefore, advanced knowledge and skills are necessary. 2) Only means to improve the recognition accuracy is tuning of the parameter. It takes long time to the system development. To solve these problems, we propose introduce the domain-specific development into the activity recognition application development. In the domain-specific development, the application is developed by using DSL (Domain-Specific Language), that is the language for a specific domain. In this paper, we present results of the design of LOARA that is DSL for activity recognition, and show evaluations.

1. はじめに

近年、無線通信機能および加速度センサをはじめとするジャイロセンサ、気圧センサ、GPS、地磁気センサなどの各種センサを有する装着型デバイスを用い、人間の行動認識を目的とした行動認識アプリケーションのための研究開発が活発に行われている¹⁾。

行動認識アプリケーションはセンサデータを取得するデバイスから行動クラスを認識するサーバまでを含め、協調して動作させなければならない。このような行動認識アプリケーションの開発上の課題として、以下の2つが挙げられる。

課題1 アプリケーション開発に必要な知識がセンサデバイスとパターン認識の2つの分野をまたいでおり、開発に高度な技術が必要となる。

課題2 認識精度向上のための決定的な方法が存在しないため、精度を向上させるにはパラメータチューニングを行いながら再実装を繰り返す必要があり工数がかかる。

これらの開発上の課題を解決することができれば、行動認識アプリケーションの開発効率を向上させることができる。

一方、ソフトウェア開発手法の1つにドメイン特化型開発²⁾がある。ドメイン特化型開発では、アプリケーション開発の際にGPL (General Purpose Language: 汎用プログラミング言語)ではなく、DSL (Domain-Specific Language: ドメイン特化型言語)と呼ばれる特定ドメインのアプリケーションを効率的に開発するための言語を開発し、開発者はそのDSLを用いてソフトウェア開発を行う。グラフィカル形式のDSLでは、直感的でわかりやすい図を記述することで動作するソフトウェアのソースコードを生成できるため、開発者は専門的な知識を持たなくともアプリケーションの開発が可能となり、開発工数を削減し、品質の高いソフトウェアを開発することができる。

そこで本研究では、行動認識アプリケーション開発にドメイン特化型開発を導入し、アプリケーション開発効率の向上をはかる。アプリケーション全体を表すモデルを記述すること

^{†1}九州大学大学院システム情報科学府

Graduate School of Information Science and Electrical Engineering, Kyushu University

^{†2}九州大学システム LSI 研究センター

System LSI Research Center, Kyushu University

^{†3}九州工業大学大学院工学研究院

Graduate School of Engineering, Kyushu Institute of Technology

^{†4}九州大学大学院システム情報科学研究院

Faculty of Information Science and Electrical Engineering, Kyushu University

で、行動認識アプリケーションを構成するために必要な、サーバからデバイス上のソフトウェアまで含めたソフトウェアを生成可能な DSL を提案する。アプリケーション開発者は提案 DSL を用いることにより、ソフトウェア開発に専門的な知識を必要とせず、行動認識アプリケーションを容易に開発・チューニング可能となり、前述した課題の解決ができると考える。

本論文の構成を以下に示す。まず 2 節でドメイン特化型開発および行動認識アプリケーションについて述べる。次に 3 節で DSL の設計手順について説明し、4 節で提案 DSL の設計について述べる。そして 5 節では提案 DSL の評価を行い、最後に 6 節でまとめを述べる。

2. ドメイン特化型開発と行動認識アプリケーション

本節では、ドメイン特化型開発とその開発に用いる DSL、および行動認識アプリケーションについて述べる。

2.1 ドメイン特化型開発と DSL

ドメイン特化型開発²⁾はソフトウェア開発方法論の一つで、ソフトウェア開発の際に特定目的向けの言語を設計することで、ソフトウェア開発の効率を向上させる開発手法である。

ドメイン特化型開発では、DSL と呼ぶアプリケーションを効率的に開発するための言語を開発し、その DSL を用いてアプリケーションを開発する。DSL は特定ドメインでの問題解決のみを想定しており、汎用プログラミング言語 (GPL) と呼ばれる C 言語や Java などとは対照的な言語である。Visser は、“DSL は形式的に定義されたシンタックスとセマンティクスを持つ、テキストもしくはグラフィカル表記された文の集まり”と定義している³⁾。

多くの DSL はバイトコードや実行コードなどの生成を目的としておらず、出力結果は GPL で記述されたソースコードや、設定ファイルなど、他の様々な媒体向けに変換する。また、DSL はテキスト形式のものとグラフィカル形式のものが存在し、テキスト形式の DSL の例としては、テキスト形式のものでは構文解析プログラムを生成する yacc、RDBMS を操作する SQL、HTML やその要素を拡張する CSS、ソフトウェアを構築方法を記述する Make などが挙げられ、グラフィカル形式の DSL の例としては、ソフトウェア無線ドメイン向け DSL⁴⁾ や、無線センサネットワークのための Software Factories である Flow⁵⁾ 我々の開発した無線センサネットワークアプリケーション向け DSL である LOSA⁶⁾ などが挙げられる。

ドメイン特化型開発の利点として

- 可変部も含めたソースコードの自動生成が可能となり、開発者の生産性が向上する。
- 問題空間の観点からの取り組みが可能となり、GPL で表現した場合に生じるエラーの

範囲が縮小される。

- 理解や表現上の誤りを、開発サイクルの早い段階で発見することができる。

といったことが挙げられる。一方、欠点としては、

- DSL 開発のためのコストが必要となる。
- 開発するソフトウェアの性能が、ハードコーディングされたソフトウェアと比較して低くなる可能性がある。

といったことが挙げられる。

しかし、ドメイン特化型開発の導入により 500%以上もの生産性の向上も報告されており⁴⁾、適用可能なドメインが適切であり、開発者が容易に理解・記述可能な DSL を開発できれば、ドメイン特化型開発は有効な開発手法となりうると言える。

2.2 行動認識アプリケーション

ここで本研究が開発対象とする行動認識アプリケーションについて述べる。本研究での行動認識アプリケーションとは、各種センサを具備した装着型デバイスからセンサデータを取得し、そのセンサデータを用いて、人間の様々な動作を認識することを目的としたシステムを指す。具体的な例を以下に挙げる。

文献 7) では、3 軸加速度センサの値から、装着者の日常行動を認識するシステムを提案している。腰に装着したデバイスから 3 軸加速度センサの値を 14Hz で取得し、それぞれのセンサの値と、その値から算出した傾きを用いて行動認識を行っている。行動クラスとしては“座っている姿勢から立つ”、“歩く”、“走る”などの 9 つの動作を SVM (Support Vector Machine) アルゴリズムを用いて認識している。

文献 8) では、2 軸加速度センサと光センサの値から、装着者の日常行動を認識するシステムを提案している。eWatch⁹⁾ と呼ばれるデバイスを用い、加速度センサと光センサの値を 50Hz で取得し、用いる特徴量計算式やデバイスの装着場所を変化させながら行動認識精度の比較を行っている。行動クラスとしては“下る”、“上る”、“座る”、“立つ”、“歩く”、“走る”の 6 つの動作を、決定木アルゴリズム (C4.5) や k 近傍法などを用いて認識している。

文献 10) では、携帯電話が具備した 3 軸加速度センサの値から、装着者の姿勢を推定するシステムを提案している。携帯電話で 3 軸加速度センサの値を 10Hz で取得、サーバに送信し、特徴量計算処理として、それぞれの分散値、平均値、FFT のパワースペクトルの最大値、重力加速度のベクトル方向、ベクトル方向の変化を計算している。そしてセンサの装着場所も考慮しており、行動クラスとしては“座る”、“立つ”、“歩く”、“走る”という動作を、携帯電話の装着位置を“ズボンのポケット”、“胸ポケット”、“鞆の中”と変更しながらそれぞ

れ認識している。

これらの研究を比較すると、同様の行動クラスの認識を目的としているものでも、用いるセンサデバイスの種類や、特微量計算方法、推定アルゴリズムは様々であり、行動認識の精度を向上させるための、決定的な方法は定まっていないことがわかる。このことから、行動認識アプリケーションの開発効率の向上は、有用性があると考えられる。

3. DSL 開発手順

本節では DSL の開発手順について説明する。DSL は以下に示す手順に従って開発する²⁾¹¹⁾⁶⁾。

工程 1 ドメイン定義 開発する DSL が対象とするアプリケーションの範囲を定義する。

工程 2 要件整理・フィーチャモデル作成 対象ドメインの機能要件と非機能要件を整理し、フィーチャモデルと呼ぶドメインの共通部と可変部を表現するモデルを作成する。

工程 3 DSL 設計 フィーチャモデルの可変部を表現できる DSL を定義する。

以下で各工程について説明する。

3.1 ドメイン定義

この工程では、DSL が開発対象とするアプリケーションの範囲となるドメインを定義する。DSL はここで定義するドメインに特化したものとなるため、対象とするアプリケーションを過不足なく含むドメインを定義する必要がある。

開発する DSL の対象ドメインの定義が狭すぎると、DSL で開発することのできるアプリケーションのバリエーションが少なくなり、DSL の有用性が低くなる。また、逆に対象ドメインの定義が広すぎると、開発者はアプリケーション開発の際に対象ドメインのアプリケーション開発に不必要な、冗長な表現の記述を強いられる。これにより開発者の作業負荷が増え、問題としていない部分でエラーを作りこんでしまう可能性があり、コードの可読性も低下するため、アプリケーションの開発効率が下がる。

3.2 要件整理・フィーチャモデル作成

この工程では、対象ドメインに属するアプリケーションが持つ要件（フィーチャ）の整理を行う。

フィーチャとは FODA (Feature Oriented Domain Analysis: フィーチャ指向ドメイン分析)¹²⁾において、システムのユーザや開発者がとらえるシステムの機能や特徴と定義されており、アプリケーションが提供するサービスである機能要件だけでなく、性能や信頼性、拡張性、セキュリティなどの非機能要件も含めたものとなる。

まず、対象ドメインに属するアプリケーションを調査し、フィーチャを洗い出す。そして、

対象ドメインのフィーチャモデル¹²⁾¹³⁾を作成する。製品間の共通性と相違性を表現するモデルとして、現在多くのプロダクトライン開発手法が採用しているのが、FODA¹²⁾に基づくフィーチャモデルである。このフィーチャモデルを用いて、対象ドメインのアプリケーション全てに存在するフィーチャである共通部と、アプリケーションによって存在の有無が異なるフィーチャである可変部とを整理する。

3.3 DSL の設計

この工程では、先の工程で作成したフィーチャモデルに従って、DSL を設計する。具体的には、DSL を構成する以下の 3 項目を定義する。

- ドメインモデル

ドメインモデルは DSL の中心となるもので、言語によって表される概念、そのプロパティ、およびそれらのリレーションシップを定義するものである。モデルの構成要素と、その各要素間の関連を定めた、DSL モデルを記述する際の文法となる。

- 表記方法

アプリケーション開発者が DSL モデルを記述する際の表現方法を定義する必要がある。DSL を用いたアプリケーション開発がスムーズに進むように、容易に理解可能な表記方法を定義すべきである。

- 生成物

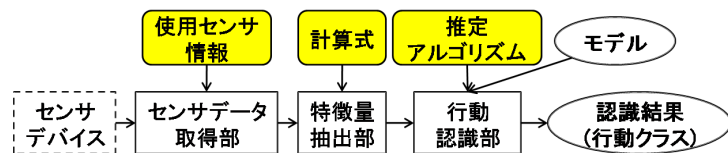
DSL の生成物と、生成物を生成するためのテンプレートを定義する。アプリケーション開発者は DSL モデルを記述し、コードジェネレータを通じて、ここで定義するテンプレートに沿った生成物を得る。テンプレートの性能が低ければ、生成物の性能も低くなるため、品質の高いテンプレートを作成する必要がある。

4. 提案 DSL

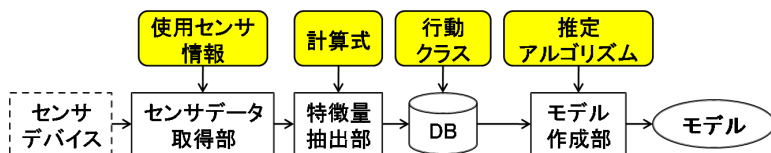
本節では、提案する行動認識アプリケーション向け DSL である LOARA (Language Of Activity Recognition Application) の設計について述べる。

4.1 対象とするドメイン

LOARA が開発対象としているドメインは行動認識アプリケーションである。具体的には以下に示す 2 つのサービスを提供するアプリケーションの開発を目的としている。行動認識 装着型デバイスからのセンサデータを用いた行動クラスのリアルタイム推定学習 行動クラス毎のセンサデータの収集、および行動認識に必要なモデルの作成 LOARA はグラフィカル形式の DSL を想定しており、アプリケーション開発者は DSL モデ



(a) 行動認識時



(b) 学習時

図 1 行動認識アプリケーションのアーキテクチャ

ルを記述することで、デバイスからサーバまでを含めた、行動認識アプリケーションを構成するために必要となるソフトウェアを一括生成できる。

4.2 要件整理・フィーチャモデル作成

LOARA が対象としているアプリケーションは、先述した 2 つのサービスを提供するものである。そこでまず、これらのサービスを実現する行動認識アプリケーションのアーキテクチャについて考察する。

行動認識アプリケーションのアーキテクチャを図 4.2 に示し、その各工程での処理内容を以下に示す。

- センサデータ取得部
 任意のセンサデバイスからセンサデータを取得する。取得したセンサデータは特徴量抽出部へ渡す。
- 特徴量抽出部

センサデータから行動コンテキスト推定のための特徴量を抽出する。特徴量の計算は、センサデータに対し、平均、分散、標準偏差を算出するものや、フーリエ変換を行うものなどがある。センサデータの生データをそのまま用いる場合もある。

- 行動認識部
 モデルと与えられた特徴量から、推定アルゴリズムを用いて行動コンテキストを認識する。推定アルゴリズムの例としては決定木や、ユークリッド距離の算出、DP マッチングなどが挙げられる。
- モデル作成部
 行動クラス毎のデータを解析し、行動認識を行うためのモデルを作成する。ここで作成するモデルは行動認識部で推定アルゴリズムとともに用いる。

日常生活での動作からジェスチャなどの特殊な動きまで、様々な行動を認識するアプリケーションが存在するが、いずれの分野のものでも処理の流れは変わらず図 4.2 に示すアーキテクチャで構築可能であると考えられる。各処理に必要な情報である、“使用センサ情報”、特徴量の“計算式”、“推定アルゴリズム”、および“行動クラス”が行動認識アプリケーションの可変部となる。

この結果を考慮し、フィーチャの整理を行い、記述したフィーチャモデルを図 2 に示す。図 2 のフィーチャモデルは FORM (Feature Oriented Reuse Method)¹³⁾ の記法に沿って記述しており、性質に基づいてフィーチャの階層分けを行ったモデルである。図 2 のフィーチャモデルが示す可変部を表現するドメインモデルを定義することで、DSL は対象ドメインのアプリケーション開発が可能な記述力を持つと考えられる。

4.3 DSL モデルの記述方法

図 2 のフィーチャモデルをもとに、ドメインモデルおよび表記方法を定義した。DSL モデルの記述方法を以下に示す。

各可変部の特性を考慮して、“SensorDeviceClass”、“FeatureQuantityClass”、“Algorithm-Class”、および“ContextGroupClass”の 4 つのクラスと、それらのクラス間の関係を記述することで、当該アプリケーションを記述するドメインモデルを設計する。クラス間の関係は結線で表す。アプリケーション開発者は、このドメインモデルに従った DSL モデルを記述していくことで、行動認識アプリケーションを開発することができる。LOARA の DSL モデルの記述例を図 3 に示す。以下では、各クラスの設定項目と DSL モデルでの表記について説明する。

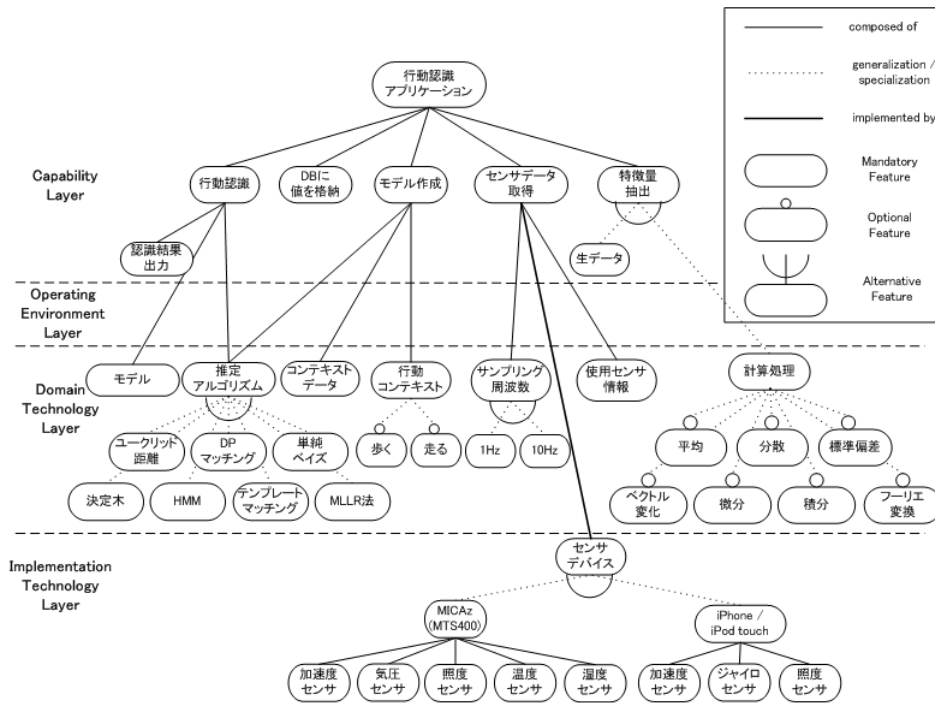


図2 行動認識アプリケーションのフィーチャモデル

4.3.1 SensorDeviceClass

SensorDeviceClass は無線センサネットワークデバイスが具備しているセンサデバイスからの、センサデータの読み出しを表現する。DSL モデルでは図3のAで表す。設定項目を以下に示す。

Name センサデータの名前を設定する。

Type SensorDeviceType (表1) から、センサデータを取得する任意のセンサデバイスを選択して設定する。

SamplingRate センサデータを取得するサンプリング周波数を Hz で設定する。

SensorDeviceClass は、1つ以上の FeatureQuantityClass と結ぶ。

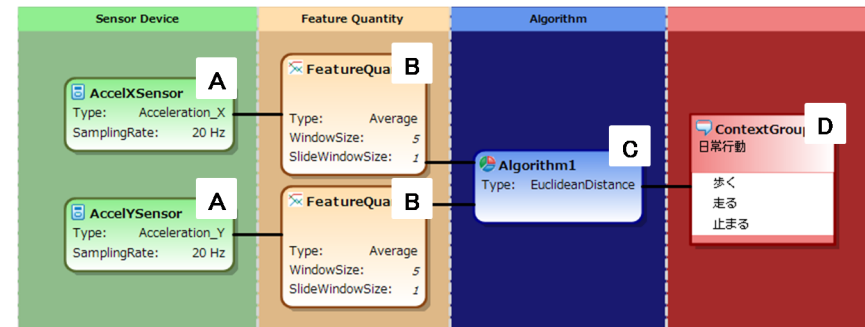


図3 LOARA のDSL モデルの記述例

表1 SensorDeviceType

変数名	使用するセンサデバイス
Acceleration.X	X 軸加速度センサ
Acceleration.Y	Y 軸加速度センサ
Acceleration.Z	Z 軸加速度センサ
GPS	GPS
Humidity	湿度センサ
Light	照度センサ
Pressure	気圧センサ
Temperature	温度センサ

表2 FormulaType

変数名	計算処理
Average	平均
Differential	微分
FourierTransform	フーリエ変換
Integration	積分
RawData	生データを用いる
StandardDeviation	標準偏差
Variance	分散
Vector	ベクトル

4.3.2 FeatureQuantityClass

FeatureQuantityClass は、センサデータからの特徴量算出処理を表現する。任意の算出式と付随するパラメータを設定し、センサデバイスから得られるセンサデータの特徴量を算出する。DSL モデルでは図3のBで表す。設定項目を以下に示す。

Name 特徴量の名前を設定する。

Type FormulaType (表2) から、任意の計算処理を選択して設定する。

WindowSize 必要である場合、特徴量を算出するために使用するデータの個数を設定する。

SlideWindowSize 必要である場合、特徴量を算出するために使用するデータのずらし幅を設定する。

FeatureQuantityClass は、1つ以上の SensorDeviceClass と結ばれており、1つ以上の FeatureQuantityClass または AlgorithmClass と結ぶ。

表 3 Algorithm Type

変数名	推定アルゴリズム
C4.5	決定木 (C4.5)
DPMatching	DP マッチング
EuclideanDistance	ユークリッド距離の比較
HMM	隠れマルコフモデル
MAP	MAP 推定
MLLR	MLLR 法
SVM	サポートベクタマシン
TemplateMatching	テンプレートマッチング

表 4 PlatformType

変数名	プラットフォーム
iPhone_iPodtouch	iPhone ¹⁴⁾ および iPod touch ¹⁵⁾
MICAz	MICAz(MPR2600J/MTS400) ¹⁶⁾

4.3.3 AlgorithmClass

AlgorithmClass は、行動認識に用いる推定アルゴリズムを表現する。任意の推定アルゴリズムと付随するパラメータを設定し、特徴量を用いて行動クラスを推定する。DSL モデルでは図 3 の C で表す。設定項目を以下に示す。

Name アルゴリズムの名前を設定する。

Type AlgorithmType (表 3) から、任意の推定アルゴリズムを選択して設定する。

Size 必要である場合、行動認識を行うために使用するデータの個数を設定する。

AlgorithmClass は、1 つ以上の FeatureQuantityClass から結ばれており、1 つ以上の ContextGroupClass と結ぶ。

4.3.4 ContextGroupClass

ContextGroupClass は認識する行動クラス群を表現する。行動クラスを複数設定し、推定アルゴリズムを用いてモデルとの比較を行い、最も結果が良い行動クラスを行動認識結果として出力する。DSL モデルでは図 3 の D で表す。設定項目を以下に示す。

Name 行動クラス群の名前を設定する。

Description 行動クラス群の説明を設定する (任意)。

Context 行動クラスを設定する。Context は複数設定することができる。Context は Name と Description を設定項目として持ち、行動クラス毎に名前と行動の説明を設定できる (Description の設定は任意)。

ContextGroupClass は、1 つの AlgorithmClass から結ばれている。

4.3.5 モデル全体の設定項目

前述した 4 つのクラス以外の設定項目として、モデル全体の設定項目が存在する。設定項目を以下に示す。

Name 開発するアプリケーションの名前を設定する。

Description 開発するアプリケーションの説明を設定する (任意)。

Platform PlatformType (表 4) から、開発するアプリケーションを動作させるデバイスのプラットフォームを設定する。

4.4 ネットワークアーキテクチャの定義

我々は無線センサネットワークアプリケーション全体を一括して生成する DSL を構想としており、既に無線センサネットワークアプリケーションのネットワークアーキテクチャレイヤのソフトウェアを生成する DSL である LOSA (Language Of wireless Sensor network Application)⁶⁾を開発している。本論文で提案する LOARA は、LOSA の上位レイヤに位置する DSL となる。LOARA の DSL モデルでは、ネットワークアーキテクチャをはじめとする基本的な無線センサネットワークの動作は、共通部となるため自動生成し記述しない。しかし、開発者がネットワークアーキテクチャに関する動作を定義する場合、LOARA の DSL モデルを LOSA が解釈できるモデルに変換し、組み合わせることで定義できる。

5. 実装と評価

本節では、LOARA を実装し、その評価を行った結果について述べる。

5.1 LOARA の実装

今回、提案 DSL である LOARA を、Microsoft DSL Tools¹⁷⁾を用いて実装した。LOARA のアプリケーション開発画面を図 4 に示す。今回実装したテンプレートは全体で 4000 行程度であるが、ドメインモデルの全ての可変部のテンプレートを実装しているわけではない。

アプリケーション開発者は、ツールボックス (図 4 の A) から任意のクラス、または関係を表すコネクタを選択し DSL モデル (図 4 の B) 上に Drag&Drop し、DSL モデルを記述していく。各クラスの項目は、任意のクラスを選択し、プロパティ (図 4 の C) から設定する。DSL モデルの記述が完了したら、“すべてのテンプレートを変換”ボタン (図 4 の D) をクリックする。DSL モデルの記述に誤りがなければ、選択したプラットフォームに応じて、デバイスからサーバまでの行動認識アプリケーションを構成するために必要なソフトウェアが一括生成される。サーバで動作するソフトウェアを図 5 に示す。

5.2 ユーザの評価

ここでは、LOARA の評価を行った結果について述べる。

被験者は、4 名の行動認識アプリケーション開発経験者および 8 名の行動認識アプリケーション開発未経験者、計 12 名である。まず被験者に行動認識アプリケーションおよび LOARA の記述方法について 10 分程度の説明を行い、被験者は実際に LOARA を使用、最後にアン

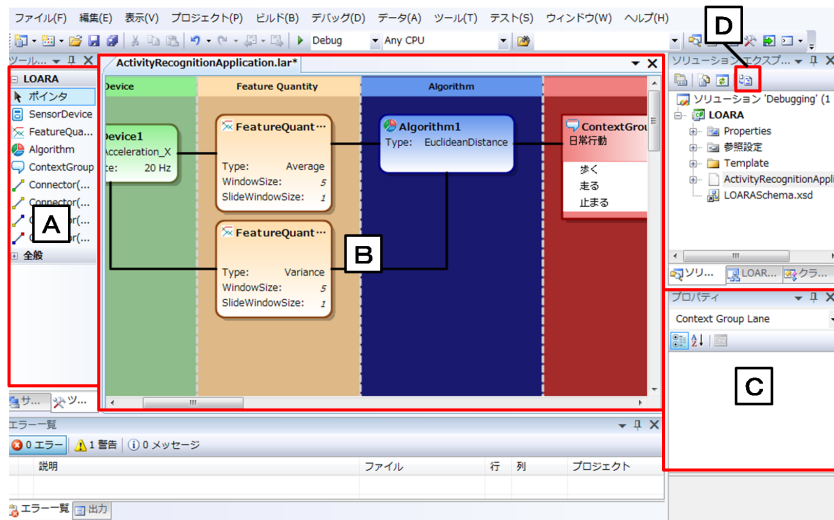


図4 LOARA のアプリケーション開発画面

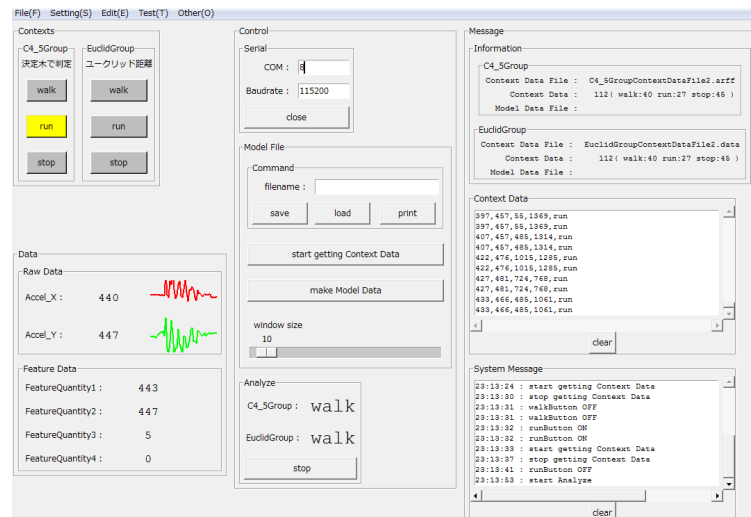


図5 生成ソフトウェア (サーバ)

表5 評価の平均値

項目	行動認識アプリケーション開発経験者		行動認識アプリケーション開発未経験者	
	既存手法	LOARA	既存手法	LOARA
生産性	3.0	4.5	2.5	4.6
学習	2.5	4.5	2.5	5.0
可読性	2.5	5.0	2.0	4.4
汎用性	4.0	3.0	4.1	2.8
品質	3.3	2.3	2.8	3.4
平均	3.1	3.9	2.8	4.0

シートに回答する形で評価を行った。

被験者が回答するアンケートの項目を以下に示す。各項目5段階評価(5:強く思う〜3:どちらともいえない〜1:全く思わない)で、既存手法(GPL)での開発とLOARAでの開発について、それぞれ回答してもらった。

生産性 アプリケーション開発の生産性が高い。

学習 記述方法を容易に学習できる。

可読性 アプリケーションの動作を理解しやすく、容易に改造可能である。

汎用性 目的分野で開発可能なアプリケーションの範囲が広い。

品質 生成したソースコードの品質が高い。

各項目の評価の平均値を表5に示す。

行動認識アプリケーション開発経験者・未経験者ともに、“生産性”、“学習”、“可読性”の項目については既存手法と比較してLOARAの方が評価が高くなった。実際に、行動認識アプリケーション開発未経験者でも、LOARAを用いることで要件通りのアプリケーションを開発・改造することができた。

しかし、“汎用性”の項目については、行動認識アプリケーション開発経験者・未経験者ともにLOARAの方が評価が低くなった。LOARAでは、行動認識アプリケーション開発の難しさを隠すため、開発者は特徴量抽出の計算処理や推定アルゴリズムなどを前述した選択範囲の中から選択する。各設定項目の選択範囲については、行動認識アプリケーションに用いられているものを全て登録することは現実的に困難であるため、行動認識アプリケーションにおいて頻繁に用いられているものを選出し、あらかじめ変数として登録している。しかし、行動認識ドメインの全ての選択項目を網羅出来ていないため、目的のアルゴリズムなどが選択できないといった場面が想定される結果であると考えられる。

そして、“品質”の項目については、行動認識アプリケーション開発経験者と未経験者に

よってその評価が分かれた。未経験者は、行動認識アプリケーションを開発したことがないため、自分で汎用言語を用いてアプリケーションを開発するよりも、LOARAを用いてアプリケーションを生成した場合の方が高い品質のアプリケーションが生成できると感じた結果であると考えられる。

全体の平均値は、行動認識アプリケーション開発経験者・未経験者ともにLOARAの方が評価が高くなった。

今後、“汎用性”と“品質”の向上のために、LOARAを拡張していく必要があるが、各項目同士が直交していないことも考慮する必要がある。例えば、記述可能なクラスや設定項目を増やすことで、“汎用性”は向上すると考えられるが、DSLモデルで記述しなければならない項目も増えるため、“学習”や“可読性”は低下すると考えられる。このような他項目への影響も考慮して、今後のLOARAの拡張計画を定めていく必要があると考えている。

6. おわりに

本研究では行動認識アプリケーション向けドメイン特化型開発言語であるLOARAを提案した。開発者はLOARAを用いることで、アプリケーション開発の際に当該ドメインの可変部のみを記述すれば良くなるため、専門的な知識を必要とせず効率的に行動認識アプリケーションを開発できる。LOARAはグラフィカル形式のDSLであり、4つのクラスとその関係を記述することで、行動認識アプリケーション全体を表すDSLモデルを記述するドメインモデルを設計した。そして、LOARAをMicrosoft DSL Toolsにて実装し、評価を行った。12名のユーザによるアンケート調査の結果、“生産性”、“学習”、“可読性”の項目で既存手法よりもLOARAの方が評価が高くなった。

今後の課題としては、LOARAの“汎用性”を向上させるため、特徴量計算式や推定アルゴリズムの新規選択項目を追加するためのDSLの拡張を行うことや、可変部となっている項目を再検討し、ドメインモデルを改良していくことが挙げられる。

謝辞 本研究は科学研究費補助金特定領域研究情報爆発IT基盤(21013038)による助成を受けている。

参 考 文 献

- 1) HASC: Human Activity Sensing Consortium, <http://hasc.jp/>.
- 2) Cook, S., Jones, G., Kent, S. and Wills, A.C.: *Domain Specific Development with Visual Studio DSL Tools (Microsoft .Net Development)*, Addison Wesley (2007).

- 3) Visser, E.: WebDSL: A case study in domain-specific language engineering, *Generative and Transformational Techniques in Software Engineering (GTTSE 2007), Lecture Notes in Computer Science*. Springer, Vol.5235, pp.291–373 (2008).
- 4) Bhanot, V., Paniscotti, D., Roman, A. and Trask, B.: Using domain-specific modeling to develop software defined radio components and applications, Proc. of the 5th OOPSLA Workshop on Domain-Specific Modeling (DSM '05), San Diego, California, USA (2005).
- 5) Naumowicz, T., Schröter, B. and Schiller, J.: Prototyping a software factory for wireless sensor networks, *Proc. of the 7th ACM Conference on Embedded Networked Sensor Systems*, ACM, pp.369–370 (2009).
- 6) 一野浩太郎, 久住憲嗣, 井上創造, 中西恒夫, 福田 晃: センサネットワーク向けドメイン特化型言語の提案, マルチメディア分散協調とモバイル (DICOMO2009) シンポジウム論文集, pp.1578 – 1587 (2009).
- 7) Song, S., Jang, J. and Park, S.: An Efficient Method for Activity Recognition of the Elderly Using Tilt Signals of Tri-axial Acceleration Sensor, *Lecture Notes in Computer Science*, Vol.5120, pp.99–104 (2008).
- 8) Maurer, U., Smailagic, A., Siewiorek, D. and Deisher, M.: Activity recognition and monitoring using multiple sensors on different body positions, *Proc. of the Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006)*, Vol.5847, pp.113–116 (2006).
- 9) Maurer, U., Rowe, A., Smailagic, A. and Siewiorek, D.: eWatch: a wearable sensor and notification platform, *Body Sensor Networks Workshop, submitted*, Citeseer, pp.142–145 (2006).
- 10) 倉沢 央, 川原圭博, 森川博之, 青山友紀: センサ装着場所を考慮した3軸加速度センサを用いた姿勢推定手法, 情報処理学会研究報告 (ユビキタスコンピューティングシステム研究会), No.54, pp.15–22 (2006).
- 11) Mernik, M., Heering, J. and Sloane, A.: When and how to develop domain-specific languages, *ACM Computing Surveys (CSUR)*, Vol.37, No.4, pp.316–344 (2005).
- 12) Kang, K., Cohen, S., Hess, J., Novak, W. and Peterson, A.: Feature-oriented domain analysis (FODA) feasibility study, *Technical Report SEI/CMU-90TR-21, Software Engineering Institute, Carnegie Mellon University* (1990).
- 13) Kang, K., Kim, S., Lee, J., Kim, K., Shin, E. and Huh, M.: FORM: A feature-oriented reuse method with domain-specific reference architectures, *Annals of Software Engineering*, Vol.5, No.1, pp.143–168 (1998).
- 14) iPhone: <http://www.apple.com/iphone/>.
- 15) iPod touch: <http://www.apple.com/ipodtouch/>.
- 16) Crossbow: <http://www.xbow.com/>.
- 17) Microsoft: Microsoft Domain-Specific Language Tools, <http://www.domainspecificdevelopment.com/>.