

# 行動履歴蓄積方法に関する研究

2005MT074 永井 有希

指導教員 沢田 篤史

## 1 はじめに

近年、ホームネットワークシステム (HNS) により個人の趣味・趣向に合わせた利用者支援を実現することが求められている。このような利用者支援に、利用者の行動履歴を活用することが考えられている [1]。利用者の行動履歴は、生活環境に遍在する情報家電や各種センサにより取得できる。様々に取得される行動履歴を、いかに整理された構造で関連付け、保守するかが重要となる。

従来の研究の多くは、行動履歴のデータ構造設計が場当たり的である。行動履歴のデータモデルを作成していない場合が多く、データ間の関係の把握が困難である。それゆえ HNS を構成する機器の変更に対し、データ構造の変更箇所が明確に分からないという問題がある。

本研究の目的は、行動履歴のデータモデルを作成する方法を提案することである。この方法では、複数の異なる視点を持つモデルから行動履歴のデータモデルを合成する。これにより、HNS を構成する機器の変更に対して履歴管理者が行なう変更箇所が明確になり、変更箇所が少なくなる。

## 2 HNS における行動データモデリング

利用者を取りまく状況の情報をコンテキストと呼び、コンテキストに応じて制御を変えるアプリケーションをコンテキストウェアアプリケーションと呼ぶ。

観測可能な利用者の行動に、機器に対する操作がある。HNS におけるコンテキストウェアアプリケーションには、利用者の機器に対する操作の履歴を用いて利用者支援を行なうものがある。このようなコンテキストウェアアプリケーションを分析した結果、行動の要素として次のものを抽出した。

- 行動の主体
- 行動の対象
- 行動の内容
- 主体が存在する環境の情報

この要素をもとに、データモデルを作成する。

## 3 行動履歴のデータモデル作成方法の提案

### 3.1 データモデルを作成する手順

本研究で提案する行動履歴のデータモデルを作成する方法を説明する。

本研究で提案する方法は次の手順で行なう。

1. HNS を構成する機器のモデルを作成
2. コンテキストとしての外部環境の要素と機器との関係を示すモデルを作成
3. 利用者のモデル作成。
4. 上記 3. 4. で作成したモデルを用いて、変換規則からデータモデルを生成。

モデル化を行なう際に、その視点を機器、外部環境、利用者分割する。このようにすることで、オブジェクト

指向により自然なモデル化が行なえる。モデル表記には UML を用いる。作成するモデルには UML プロファイルを定義する。UML プロファイルを用いることで、モデル要素の付加的な意味と、意味的な制約を明示することができる。

### 3.2 モデルのための UML プロファイル

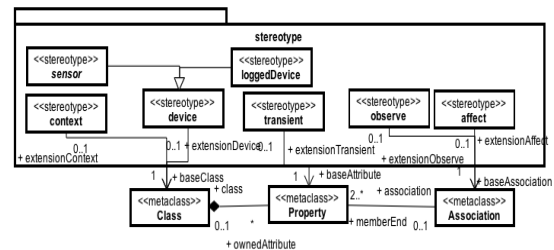


図 1 機器と外部環境のモデルを記述するための拡張 UML メタモデル

機器と外部環境の要素を記述するための UML プロファイルを定義した。このプロファイルにより拡張した UML メタモデルを図 1 に示す。<< device >> は HNS を構成する機器を示すステレオタイプである。ステレオタイプ << loggedDevice >> は、そのクラスのインスタンスが操作された際に履歴がとられる機器を示す。<< sensor >> は外部環境を計測するセンサであり、<< context >> は、コンテキストとして抽出した外部環境の要素である。値が頻繁に変わりうる属性には << transient >> が付加される。<< observe >> はセンサが外部環境の要素を計測する関係を示す。また << affect >> は外部環境の要素とそれをコンテキストとする機器の関係を示す。

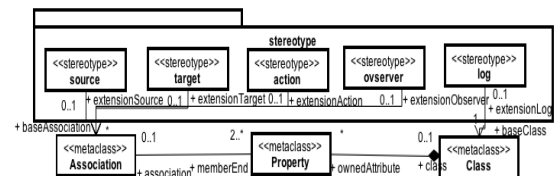


図 2 行動履歴のモデルを記述するための拡張 UML メタモデル

次に、モデル変換により生成される行動履歴のモデルに適用する UML プロファイルの説明を行う。2 節であげた行動の要素をもとに UML プロファイルを作成した。行動履歴のための拡張 UML メタモデルを図 2 に示す。ステレオタイプ << log >> は行動履歴としてまとめるクラスを示す。<< source >> は << log >> クラスと行動の主体の関係であり、<< target >> は << log >> クラスと行動の対象の関係である。<< action >> は << log >> クラスと行動の内容の関係を示すステレオタイプであ

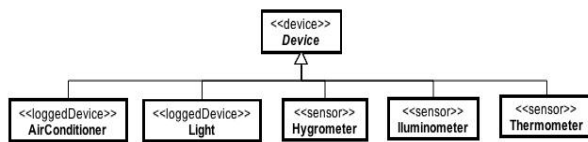


図3 HNSを構成する機器のモデル例

る。そして `<<observer>>` は `<<log>>` クラスと外部環境の要素を計測する機器の関係を示す。

#### 4 事例検証

本研究で提案する方法を、事例を示して説明する。利用者の情報家電に対する操作の最小化を目的としたコンテキストウェアアプリケーションを想定し、データモデルを作成する。

文献 [1] で適用している環境をもとに、HNS には照明、エアコン、照度計、湿度計、温度計がつながっていることを想定する。さらにこれらの機器が1つずつ、同じ部屋に収容されていることとする。

##### 4.1 機器のモデル

まず、HNS を構成する機器のモデルを作成する。図 1 で示した UML プロファイルを適用する。この図を図 3 に示す。AirConditioner, Light クラスは `<<log>>` が付加され、メソッドが呼ばればそれが履歴として残される。Hygrometer, Illuminometer, Thermometer は `<<sensor>>` が付加され、利用者が操作を行なった際の外部環境の状態を測定するためのセンサであることを示す。

##### 4.2 外部環境の要素と機器との関係を示すモデルを作成

コンテキストとして、外部環境の要素を抽出する。その後、機器との関連をモデルに記述する。図 1 で示した UML プロファイルを適用する。図 4 にこの図を示す。

AirConditioner への操作が起こる際に、湿度と温度が履歴に保存される。これらを計測するのがそれぞれ Hygrometer, Thermometer である。照明への操作が起こる際には、照度が履歴に保存される。照度を計測するのが Illuminometer である。

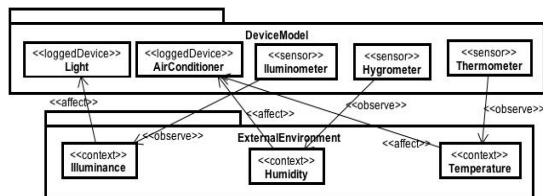


図4 外部環境の要素と機器との関連を示すモデル例

##### 4.3 データモデル例

図 4 と利用者を表すクラス図から変換規則により生成したエアコンへの行動履歴のデータモデルを図 5 に示す。

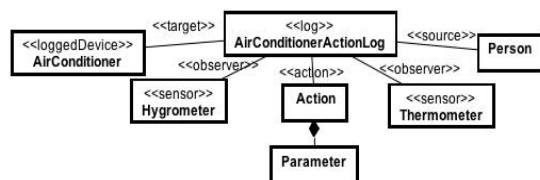


図5 AirConditionerのデータモデル

## 5 考察

本研究で提案した方法を考察する。

4 節で想定したコンテキストウェアアプリケーションを cogma[1] で実現した場合のデータ構造設計と比較する。cogma は機器間の連携アプリケーションを構築するミドルウェアである。このアプリケーションは、個々の機器への操作が行なわれる度にその状況を XML データベースへ保存する。cogma におけるデータ構造設計は、データモデルがつけられていない。操作とそれに関係する情報を組にして蓄積するという方針が決まっているのみである。場当たりにセンサの値を収集し、蓄積している。センサなどの機器の変更に対し、影響を受けるデータ構造の把握が困難である。

本研究では一連の手順によりデータモデルを作成する方法を提案した。メタモデルと変換規則を定義し、変換ツールによりデータモデルへ変換する。HNS を構成する機器に変更があった場合には、メタモデルに基づき作成したモデルから、変更に対して影響を受ける箇所の把握が容易におこなえる。機器と外部環境の要因との関係を示すモデルを修正し、このモデルを変換すれば、容易にデータモデルを修正することが可能となる。

MyLifeBits[2] は、PC を使用する際に行われる全ての操作・動作を記録するアプリケーションである。このデータモデルは、インスタンスレベルのデータがタイプをもつリンクで互いにつながり合う構造になっている。複数の視点が混在しており、複雑でわかりにくい。本研究では利用者の行動として情報家電の操作を対象としたが、PC 操作も基本的に同じである。本研究で提案する方法は、視点をわけたモデルを合成するという手法をとっている。視点を分離することでモデルがわかりやすくなり、モデルの設計や変更が容易になるという利点がある。

## 6 おわりに

本研究では、行動履歴のデータモデルを作成する方法を提案した。行動履歴の要素に視点を機器、外部環境、利用者としてモデル化する。そしてモデルを組み合わせて行動履歴のデータモデルへと変換する。また、モデルに対して UML プロファイルを定義し、モデルの意味と制約を明示した。

今後は、OCL による制約の記述とデータモデルを自動生成するツールの実現が課題として残っている。

## 参考文献

- [1] 河川 伸夫, 宮崎 俊和, 稲垣 康善, “ユビキタス情報環境における履歴を用いた機器操作支援手法,” 情報処理学会研究報告 UBI, vol. 2004, no. 39, 2004, pp. 57-62.
- [2] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong, “MyLifeBits: a personal database for everything,” *Communications of ACM*, vol. 49, issue 1, 2006, pp. 88-95.