

コンテキストウェアな組み込みソフトウェアの開発支援に関する研究

2012SE046 堀貴成 2012SE136 町野健太

指導教員：沢田篤史

1 はじめに

近年，センシングデバイスの高機能化や高性能化によってユーザやものの状況や周りの環境などのコンテキスト情報を取得できるようになった．それに伴ってコンテキストウェアな組み込みシステムが増えてきた．一般的にコンテキストは5W1Hに関する情報に基づいて構成される [1] が，我々はシステムの行動履歴もコンテキストとして重要であると考え．過去において行なったシステムの行動とその結果は，次に今システムが取るべき行動を決定するための重要な手がかりになりうるからである．

これまでにコンテキスト情報のうち主に5W1Hを処理するためにいくつかの組み込みソフトウェアアーキテクチャが提案されている [1][2]．これらはコンテキスト情報を処理するためのソフトウェアの基本構造を示しているが，行動履歴の取り扱いについては明らかでない．行動履歴をコンテキスト情報と捉え，それをシステムの制御に利用する．組み込みソフトウェアの開発を効率的に行なうためには行動履歴に関する情報を処理するモジュールを明確にしたアーキテクチャとそれに基づく開発支援が必要である．

本研究の目的は，コンテキストウェアな組み込みソフトウェアにおいて行動履歴を体系的に取り扱うことのできるソフトウェアの開発を支援することである．行動履歴のモデル化，行動履歴を取り扱うソフトウェアアーキテクチャの構築を行なうことでその目的の達成を目指す．

行動履歴のモデル化では，システムの内部状態を状態遷移図を用いた記述法でモデルの作成を行なう．状態遷移図は行動履歴のコンテキスト情報を取得し，行動を決定するために利用する．状態は行動を決定付ける基準であり，遷移はシステムの行動の結果を表す．

本研究で提案しているアーキテクチャは組み込みシステムで多イリアクティブなシステムの典型的な構造であるセンサアクチュエータパターン [3] に付加することを想定する．行動履歴を取り扱うためのソフトウェアアーキテクチャの構築では，大量に存在する様々なコンテキストの中から必要なコンテキストだけを選択して，アクチュエータの制御を行なう処理が重要である．本研究ではコンテキストを利用するアーキテクチャにパイプアンドフィルタアーキテクチャスタイル [3] を適用する．状態遷移図を用いて，行動履歴のコンテキスト情報を取得し，行動を決定するのでStateパターンを適用する [4]．また，アーキテクチャ構築はアスペクト指向に基づいて行なう．アスペクト指向を用いることで，コンテキストの処理をアスペクトとして分離して，モジュール化することが可能である．これにより保守性や再利用性を高めることができる．

提案手法の妥当性を検証するために，自律走行ロボット

を用いた模擬的な宅配システムを取り上げる．宅配先の在宅，不在に関する履歴をコンテキストとして，定義したアーキテクチャを基に自律走行ロボットの実装を行ない，アーキテクチャの妥当性の確認をする．

2 コンテキストウェアな組み込みシステムとその開発における課題

2.1 コンテキストウェアネス

コンテキストとは，一般的には「文脈」や「背景」を指し，ウェアネスとは「～と知ること，自覚，認識すること，意識，配慮」とされている．コンテキストウェアネスとは，「文脈で認識できる」という意味で，人やモノ，などの状況に応じて対応することができるという概念である．

2.2 センサアクチュエータパターン

センサアクチュエータパターン [4] は外界のイベントをセンサで捉え，それに基づいた処理を行ない，アクチュエータを介して外界に応答するパターン．組み込みシステムで多イリアクティブなシステムの典型的な例である．

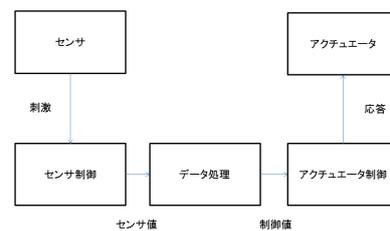


図1 センサアクチュエータ [3]

2.3 アスペクト指向

横断的関心事は，他の関心事に横断的に関連しており，単独では取り扱えない関心事である．アスペクト指向を用いることで，横断的関心事を単一モジュールとして実装することができる．アスペクト指向プログラミングの利点としては保守性や再利用性が挙げられる．

2.4 二層コンテキストモデル

Dey らの二層コンテキストモデル [1] を図2に示す．

Dey らはコンテキストを実態 (人や場所，情報，モノ) の状況 (環境や背景) を特徴付けるすべての情報であるとしている．第二層コンテキストモデルはコンテキストエンティティと個別エンティティから成り立っている．第一層がコンテキストエンティティと呼ばれるもので，状況の特徴付けると考えられている位置，時間，主体や活動の基本的な四つのエンティティから構成されている．第二層は個

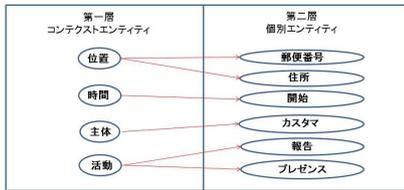


図2 二層コンテキストモデル [1]

別エンティティと呼ばれるもので、コンテキストエンティティを索引とし、対象となるドメインごとの個別属性を表すエンティティである。

2.5 三層コンテキストモデル

青山, 杉田, 中道らの三層コンテキストモデル [2] を図3に示す。

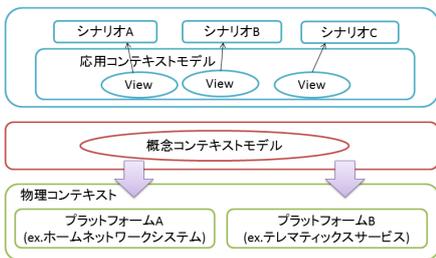


図3 三層コンテキストモデル [2]

三層コンテキストモデルは概念コンテキスト、物理コンテキスト、応用コンテキストから成り立っている。概要コンテキストは対象システムの環境下における、ユーザを含めたシステム全体を表現するコンテキストである。物事の一般的な概念である5W1Hに基づいて構成要素を抽出し、概念コンテキストモデルとして定義する。

応用コンテキストは概念コンテキストモデルに基づいて、コンテキストウェアアプリケーションを実現するシナリオを特徴付けるコンテキストである。物理コンテキストはプラットフォームに依存するコンテキストである。

Deyらの第一層のコンテキストエンティティは概念コンテキスト、第二層の個別エンティティは物理コンテキストにそれぞれ相当している。

Deyらとの違いが三層目の応用コンテキストの存在である。応用コンテキストの役割はViewによって異なるコンテキストウェアサービスに分岐させる。このモデルの課題は、特定の目的に応じてViewを抽出する方法を検討する必要がある点である。

3 行動履歴を体系的に取り扱う組込みソフトウェアの開発支援

3.1 行動履歴のモデル化

本研究ではコンテキスト情報として行動履歴を加える。過去において行なったシステムの行動とその結果は、次に今システムが取るべき行動を決定するための重要な手がかりになりうるからである。行動履歴に関するコンテキ

スのモデル化を実現するために、我々は状態遷移図を用いる。システムの内部状態を状態遷移図で表現する。状態遷移図を用いることで、行動履歴のコンテキスト情報を取得し、行動を決定するために利用する。

行動履歴と状態遷移図の関連図4に示す。

状態	行動を決定付ける基準
イベント	概念コンテキスト情報
遷移	システムの行動の結果
アクション	システムの行動

図4 行動履歴と状態遷移図の関連図

3.2 アーキテクチャスタイルの適用

3.2.1 パイプアンドフィルタ

大量にある様々なコンテキスト情報のなかで必要なコンテキスト情報だけを選択して、アクチュエータの制御を行える設計を行う必要がある。我々は本研究のアーキテクチャスタイルにパイプアンドフィルタスタイルを適用する。

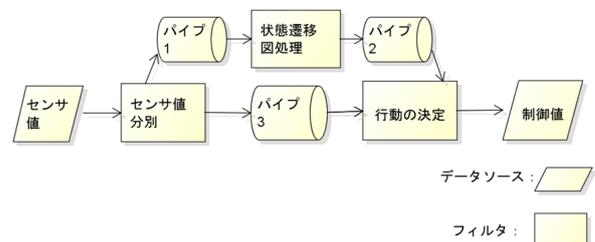


図5 提案しているアーキテクチャ

センサ値分別は物理コンテキストであるセンサ値を取得、概念コンテキストとして分別し、パイプ1、パイプ3に書き出す。状態遷移図処理はパイプ1からデータを取得、行動の結果から状態を遷移、行動履歴から状態を判断し、パイプ2に書き出す。行動の決定はパイプ2、パイプ3からデータを取得、コンテキストを判断し行動を決定、行動情報を出力する。

3.3 関心事の分離とアスペクト指向アーキテクチャの設計

行動履歴と横断する関心事の概要を図6に示す。

アーキテクチャ構築はアスペクト指向に基づいて行なう。アスペクト指向を用いることで、コンテキストの処理をアスペクトとして分離して、モジュール化することが可能である。これにより保守性や再利用性を高めることがで

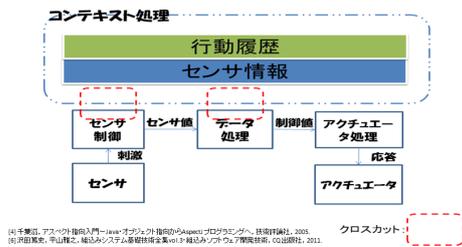


図 6 横断的関心事

きる．コンテキスト処理ではセンサ情報を取得し，行動の決定を行なうので，センサアクチュエータパターンのセンサ制御とデータ処理に関するモジュールは横断的関心事となる．

4 事例による検証

4.1 自律走行ロボットによる宅配アプリケーションのモデル化

我々は行動履歴に基づいて在宅時を狙って宅配させるアプリケーションを作成．

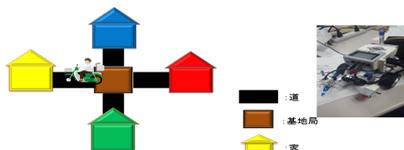


図 7 事例で用いる検証

曜日ごとにどこの家が在宅か不在かを判断させ，宅配させる．行動履歴に必要なコンテキスト情報は 5W1H に基づく．Who に関しては宅配する際に家の誰が受け取るかは重要ではないので Where と同義である．What と How に関して，宅配物を渡せたか渡せなかったかは在宅していたか不在だったかに依存しているので Why と同義である．まとめたものを図 8 に示す．

5W1H	When	Where, Who	What, How, Why
コンテキスト	時間 (午前, 午後), 曜日	家の誰かが	宅配物を渡せたか or 渡せなかったか. それは在宅だったか or 不在だったから.
	内蔵タイマー	色	タッチセンサーが押下された or されなかった
抽出方法	内蔵タイマー	カラーセンサ	タッチセンサ

図 8 検証で用いるコンテキスト情報まとめ

行動履歴のモデル化では曜日ごとの各家の在宅か不在かを状態遷移図を用いて表現する．図 9 に示す．

宅配完了か宅配未完了により状態が遷移する．連続で不在が続いた場合は宅配順序の優先度が下がる．

自律走行ロボットで検証を行う際の状態遷移図を具体例である．この状態遷移図は宅配先，曜日ごとで作成される．状態は在宅，ほぼ在宅，ほぼ不在，不在の 4 つである．イベ

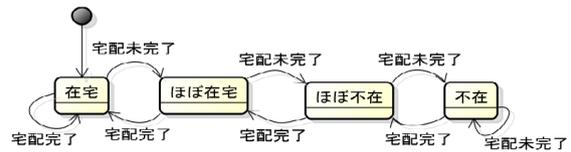


図 9 モデル化の具体例

ントは宅配完了，宅配未完了の 2 つである．在宅，ほぼ在宅の状態のときに宅配．不在，ほぼ不在のときは後回し．

1. 初期状態を在宅とする．宅配完了というイベントを得た場合，在宅の状態をループする．在宅の状態から宅配未完了というイベントを得た場合，ほぼ在宅に遷移する．
2. 状態がほぼ在宅の時は，宅配完了というイベントを得た場合，在宅に遷移する．宅配未完了というイベントを得た場合，ほぼ不在に遷移する．
3. 状態がほぼ不在の時は，宅配完了というイベントを得た場合，ほぼ在宅に遷移する．宅配未完了というイベントを得た場合，不在に遷移する．
4. 状態が不在の時は，宅配完了というイベントを得た場合，ほぼ不在に遷移する．宅配未完了というイベントを得た場合，不在の状態をループする．

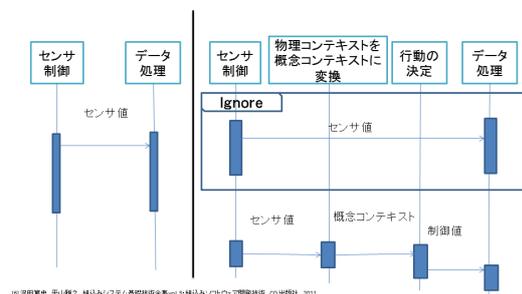


図 10 提案したソフトウェアアーキテクチャスタイルに基づいたシーケンス図その 1

センサがデータ処理にセンサ情報を送信する前後をジョインポイントとする．物理コンテキストを概念コンテキストに変換はそのジョインポイントにフックし，センサから物理コンテキストを概念コンテキストに変換にセンサ情報を送信．変換された概念コンテキストを行動の決定に送信する．行動の決定で決定された行動をデータ処理に送信．

物理コンテキストを概念コンテキストに変換から行動の決定に送信した後をジョインポイントとする．状態遷移図はそのジョインポイントにフックし，物理コンテキストを概念コンテキストに変換から概念コンテキストを状態遷移図に送信．状態遷移図で次の目的地の状態を判断し行動の決定に送信．また，状態遷移図では状態遷移図の更新も行なわれる．

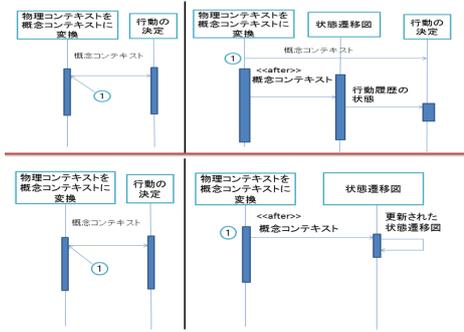


図 11 提案したソフトウェアアーキテクチャスタイルに基づいたシーケンス図その 2

4.2 目的に対してのアプローチの妥当性

本研究の目的は、コンテキストアウェアな組込みソフトウェアの開発支援することである。よって、行動履歴のモデル化をすることでコンテキスト情報と行動履歴の関連が明確になり行動履歴をどう表現するかの枠組みが与えられる。

行動履歴を取り扱うソフトウェアアーキテクチャを構築することはコンテキストアウェアネスと行動履歴の関係や位置づけの把握などの確認が行うことが可能となる。行動履歴の分析を行い、ソフトウェアアーキテクチャを構築し、一般化することにより行動履歴を体系的に取り扱うことのできるソフトウェアの開発を支援することができる。

4.3 提案しているモデル化手法の妥当性

二層コンテキストモデルでは、入力に対して、単純に出力が対応しているため、行動履歴に応じたコンテキスト情報を扱うことが難しい。行動履歴に応じたコンテキスト情報を扱うためには、別の仕組みが必要となる。そこで我々の状態遷移図を用いたモデルを用いることにより行動履歴に応じたコンテキスト情報を扱うことが可能である。このようなことから我々の行動履歴に関するモデル化は妥当であると言える。

三層コンテキストモデルでは、View の部分を状態遷移図に置き換えることにより、我々と同じ行動履歴に関するモデル化が可能となる。状態遷移図で表現していない分、我々の行動履歴に関するモデル化には意義があると言える。

4.4 提案しているアーキテクチャ手法の妥当性

実際にパイプアンドフィルタを適用したアーキテクチャを元に、自律走行ロボットで検証を行なう。行動履歴をコンテキスト情報として、仕様や要求通りにシステムが稼動することが確認をすることにより、我々のアーキテクチャの妥当性を確認することができる。

5 考察

行動履歴を取り扱うためのソフトウェアアーキテクチャの構築にあたり、他のアーキテクチャスタイルやデザイン

パターンなどを適用した場合と我々のアプローチの比較を行なった。

初めに、行動履歴のモデル化でデータベースを用いた場合と行動履歴のモデル化で状態遷移図を用いた場合の比較を行なった。データベースを用いた場合は、行動履歴に必要なコンテキスト情報を大量に蓄積する必要がある。これに対し状態遷移図を用いた場合は、行動履歴のコンテキスト情報を取得し、状態遷移図として蓄積されるので大量のデータを蓄積する必要がない。状態遷移図の問題としてあらゆる行動や状態を想定した状態遷移図を作成し、モデル化を行なうのに時間がかかることがあげられる。次に、アーキテクチャスタイルの適用において、パイプアンドフィルタスタイルとリポジトリスタイルを適用した場合とパイプアンドフィルタスタイルのみを適用した場合の比較を行なった。リポジトリスタイルを適用することによって、コンテキスト情報の保存が可能であるが、リポジトリスタイルを適用しなかった場合でも、行動履歴を状態遷移図で表現できる。リソースが厳しい組込みシステムにおいては、我々が提案しているパイプアンドフィルタスタイルのみを適用させたアーキテクチャの方が組込みシステムに適していると言える。

6 おわりに

本研究では、行動履歴に関する情報を処理するモジュールを明確にしたアーキテクチャを提案した。コンテキストアウェアな組込みソフトウェアにおいて行動履歴を体系的に取り扱うことのできるソフトウェア開発支援を行なうことができるようになった。今後の課題としては宅配システム以外を想定した検証を行なうことが挙げられる。

参考文献

- [1] Anind K. Dey, "Providing Architectural Support for Building Context-Aware Applications", <http://www.cc.gatech.edu/fce/ctk/pubs/dev-thesis.pdf>, 2000.
- [2] 青山幹夫, 杉田達哉, 中道上, "概念コンテキストモデルに基づくコンテキストアウェアサービス開発手法の提案," 情報処理学会第 37 回全国大会, 2011.
- [3] 沢田篤史, 平山雅之, 組込みシステム基礎技術全集 vol.3・組込みソフトウェア開発技術, CQ 出版社, 2011.
- [4] P. Clements, F. Bachmann, L. Bass, D. Garlen, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, *Documenting Software Architectures-Views and Beyond*, Addison-Wesley, 2011
- [5] 千葉滋, アスペクト指向入門ー Java・オブジェクト指向から AspectJ プログラミングへ, 技術評論社, 2005.