

機能協調型家電ミドルウェアへの 実機能接続を支援するツールの設計

2004MT054 前田 泰志

指導教員 蜂巢 吉成

1 はじめに

近年、家庭内の機器がホームネットワークに接続された環境が普及しつつあり、家電やセンサなどのアプリケーション（以下 NA:Networked Appliance と呼ぶ）がホームネットワーク上で互いに連携しながら家庭環境に適応したサービスを提供するためのプラットフォーム「ゆかりカーネル」が提案されている [1]。ゆかりカーネルは各種 NA を機能（以下 FE:Function Element と呼ぶ）に分割し、接続・協調させることによって、家庭環境に適応したサービスを提供するミドルウェアである。

ゆかりカーネルにおいて NA はイニシエータとレスポンドに分類される。イニシエータは連携動作が書かれたサービス記述に従い FE 発見や実行制御を行う。レスポンドはイニシエータに対する応答や、NA の持つ機能を FE として制御する。FE の仕様は NA 記述として NA 毎に記述されており、レスポンドはこの記述に従って FE の制御を行う。機能クラスとは FE 制御を行うプログラムである。FE によっては外部から受け付けられる制御コマンドが異なるので、FE 毎に機能クラスを作成する必要がある。

家電をゆかりカーネルを利用して動作させるためには、家電の持つ機能毎に機能クラスを手作業で作成する必要があり、開発にかかる手間が大きいことが問題点としてあげられる。

本研究の目的は、レスポンドの開発を支援するために、NA 記述を入力として機能クラスを自動生成する実機能接続支援ツールの作成である。異なる FE に対する機能クラスでも、受け付ける制御が同じであれば、その部分の処理は共通部分が多いので、それを定型部分として自動生成する。

2 ゆかりカーネルの概略

ゆかりカーネルの構成の概略を図 1 に示す。レスポンドは NA マネージャと FE マネージャで構成されていて、イニシエータの機能発見要求に対し NA マネージャは応答や接続を制御する。FE マネージャは家電が持つ実機能を FE として制御していて、この部分が機能クラスにあたる。

NA が持つ FE の仕様は NA 記述として XML を用いて記述され、FE の詳細仕様は FE 記述として NA 記述内に定義される。FE 記述には、FE の ID、FE 名、操作タイプ、制御特性リストなどが記述される。操作タイプは FE がデータを生成するか消費するかを定義しており、それに応じて、機能クラスにはデータ通信メソッドが必要となる。制御特性リストには FE が受け付けること

出来る制御コマンドが記述され、制御コマンドに対応して、FE 制御用のリクエスト受信処理を定義するメソッド（以後 FE 制御メソッドと呼ぶ）が必要となる。必要なメソッドは FE 毎に異なるが、同じ制御コマンドを処理するメソッドは、家電の機能呼び出ししている部分以外は FE に関わらず共通な処理（定型部分）となる。

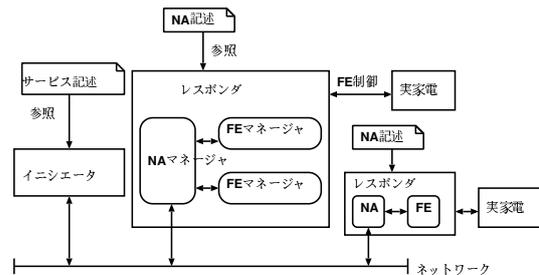


図 1 ゆかりカーネルの構成概略

3 実機能接続支援ツールの提案

家電毎に必要な機能クラスを手作業で作成しているという問題点を解決するために、本研究では NA 記述から機能クラスの定型部分を自動生成する実機能接続支援ツールを提案する。機能クラスの定型部分の全てをあらかじめ用意しておく方法もあるが FE 毎に受け付けることが出来る制御コマンドが異なるので、FE の処理に従った機能クラスをそれぞれ生成する必要がある。家電の実機能呼び出ししている部分は FE 毎に異なるので、コメントとして生成しておき、ツールの利用者が書き換える。ツールの概略を図 2 に示す。

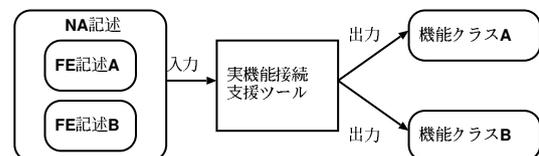


図 2 実機能接続支援ツールの概略

ツールの設計をするために、機能クラスの構成を整理し、定型部分をまとめ、FE 記述から定型部分を生成する方法を提案する。

3.1 機能クラスの構成

ツールが生成する機能クラスの構成要素には次のものがある。

3.1.1 コンストラクタ・デストラクタ

コンストラクタ・デストラクタでは機能クラスのオブジェクトの生成及び消滅時に実行する処理（変数の初期化や確保済みメモリの領域の開放など）を定義している。これらは FE に関わらず共通処理なので定型部分として用意出来る。

3.1.2 FE 制御メソッド

FE 制御メソッドは、イニシエータからの制御コマンドに対して処理を行うメソッドである。家電の実機能呼び出ししている部分以外は共通な処理である。図 3 は FE 制御メソッドの中の 1 つである comStart() の処理の例である。点線で囲まれたデバイスの初期化処理以外は共通な処理なので定型部分となる。

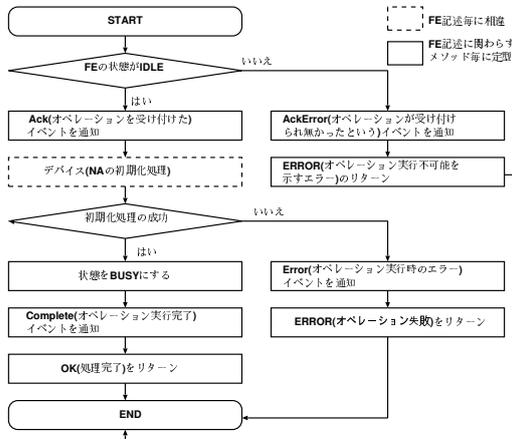


図 3 comStart() の処理

3.1.3 データ通信メソッド

データ通信メソッドは家電の連携において、FE 間でデータを送受信するメソッドである。データ通信メソッドには、イニシエータからの処理要求に対してデータ送信などの処理を行うメソッドであるデータ通信リクエスト受信処理メソッドと、他の FE からデータを受信した際に呼び出されるメソッドであるデータ受信コールバックメソッドの 2 種類がある。

イニシエータに対するイベント通知及び FE 間でのデータ通信部分は共通の処理なので定型部分となる。家電からデータを読み出して来る部分と家電にデータを送信する部分は家電毎に異なる。FE 制御メソッドやデータ通信リクエスト受信処理メソッド、データ受信コールバックメソッドは全て定型部分として扱う事が出来る。各メソッドの定型部分と、FE 記述毎に相違な部分をフローチャート図でまとめた。図 3 は FE 制御メソッドの 1 つである comStart() を例にあげた。

3.2 実機能接続支援ツールの設計

NA 記述から機能クラスを自動生成するために必要な対応関係を提案する。

3.2.1 クラス名

図 4 に示すように、機能クラス名およびファイル名は FE の名前から生成される。

3.2.2 コストラクタ及びデストラクタ

コンストラクタ名、デストラクタ名は FE 名から生成する。コンストラクタ及びデストラクタ内の定型部分はあらかじめ用意しておく、FE に関わらず全ての機能クラスに生成する。

3.2.3 FE 制御メソッド

FE 記述内の制御特性リスト (ControlProperty List) に記述された FE が受け付ける制御コマンドに対応して、

FE 制御メソッドの定型部分を生成する。図 4 ではリストに start, stop が定義されているので、機能クラスには comStart(), comStop() が生成される。

3.2.4 データ通信メソッド

FE 記述内の操作タイプ (OperationType) によってデータ通信メソッドを生成する。図 4 のでは、操作タイプが Generate なので comSendCSP() を生成する。操作タイプが Consume の場合は onReceiveData() を生成する。

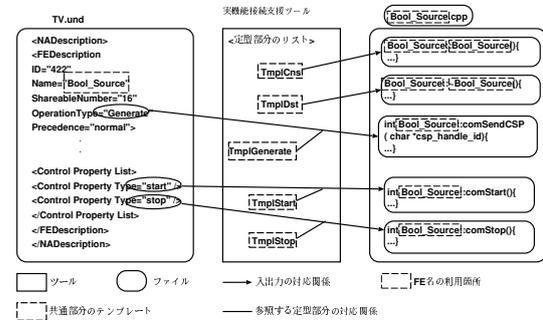


図 4 ツールの入出力の詳細

4 考察

本研究で提案する実機能接続支援ツールによって、これまで手作業で記述されていた機能クラスの定型部分が自動生成出来る。また、生成された機能クラス内で家電の機能呼び出す部分はコメントとして示されているので、ツール利用者は変更箇所を容易に特定出来る。このツールによってレスポンドの開発が従来よりも簡略化出来ると考えられる。

家電接続処理部分については、コメントの代わりに空のメソッドとその呼び出しを生成し、ツール利用者がサブクラスでそのメソッドの処理内容を記述する方法も考えられる。この場合、生成された機能クラスは変更する必要がなくなり、より容易にレスポンドの開発が出来ると考えられる。しかし、生成される空のメソッドは回数や振り値などを家電に関わらず一般化する必要がある。

5 おわりに

本研究では、機能クラスの定型部分を調べ、FE 記述と機能クラスの各構成要素の対応関係を考察し、実機能接続支援ツールの提案をした。

データ通信メソッドは comSendCSP() のみを扱ったが、他にもデータ通信メソッドはあるので、その対応関係も考える必要がある。今後の課題として実機能接続支援ツールの実現があげられる。

参考文献

- [1] 沢田 篤史: “ホームネットワーク向け機能協調基盤ゆかりカーネル”, 第 3 回ユビキタスホームワークショップ論文集, pp.111-116, 2006 .