

機能協調型家電アプリケーションのための サービス記述開発支援ツール

2004MT095 篠田 晃平 2004MT103 鈴木 豊大
指導教員 沢田 篤史

1 はじめに

多様なデジタル家電や高速ネットワーク技術の一般化により、家庭内の多種多様な機器がホームネットワークに接続された環境が普及することが見込まれている。家電などのアプライアンス (NA: Networked Appliance) 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として「ゆかりコア」が構築された [1]。さらに、ゆかりコアの提供する機能協調基盤の上で、より柔軟にユーザ対応できるサービスアプリケーションを可能とするための機能を追加したミドルウェア「ゆかりカーネル」が開発された [2]。

ゆかりカーネルでは、機能 (FE: Function Element) 間の接続関係記述をトポロジ記述と呼び、サービスアプリケーションの振舞いを形式的に表現する記述をシナリオ記述と呼ぶ。

ゆかりカーネルにおけるサービス開発の問題点として、膨大な記述量となるシナリオ記述の作成が困難であることが挙げられる。加えて、シナリオがパターン化されているにも関わらず、毎回記述しなければならない点も開発者にとって負担になっている。

本研究の目的は、ユーザのサービス開発を支援することである。その方法として、通信パターンや使用オペレーション、機能間の接続関係の観点からシナリオ記述を自動生成する、サービス記述開発支援ツールを作成する方法を提案する。

本研究は以下のように進める。

- シナリオ記述の分析
- 拡張トポロジ記述の考案
- 自動生成ツールの設計

2 研究の背景と目的

2.1 ゆかりカーネル

ゆかりカーネルは、情報家電、センサ、ロボットなどがその機能を協調させてサービスを提供するプラットフォームである。機能を機器の枠を越えて連携させることにより生まれる利点としては、以下の事が挙げられる。

- 機能の重複を省く
- 単体では実現不可能なサービスを実現
- 故障中の機能を他の機器で代替

図1は生成と消費のFEの一覧である。他にも操作方法として合成、変換、蓄積があるが、本論文ではもっとも

頻繁に使われている生成と消費のFEを中心に扱う。

	生成	消費
ブリーチン	Bg	Bc
数値	Ng	Nc
文字	Tg	Tc
音声	Ag	Ac
画像	Ig	Ic
映像	Mg	Mc

図1 FE一覧

2.2 家庭内ホームネットワークのモデル

図2は一般家庭をモデルとして、実際の家電の連携を示している。機能の分類や連携の仕様などはゆかりカーネルに基づいて記述している。

以下はゆかりカーネルにおける家電の連携の利点から考えた接続例である。

接続例1(機能の重複を省く)

- (1) 洗面所の洗濯機の終了の合図をリビングルームのラジカセで鳴らす
- (2) 大きい音を洗濯機で出す必要が無くなる

接続例2(故障中の機能を他の機器で代替)

- (1) 洗面所の洗濯機の終了の合図をリビングルームのラジカセで鳴らす
- (2) ラジカセが故障中の場合がある
- (3) 代わりにリビングルームのテレビで音を鳴らす

接続例3(単体では実現不可能なサービスを実現)

- (1) 外のカメラで外の様子を映す
- (2) テレビにその様子を映す
- (3) 家の中でも外の様子がわかる

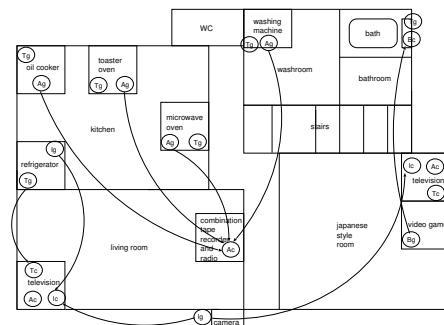


図2 家庭内ホームネットワークの一例

2.3 シナリオ記述

ゆかりカーネルにおけるサービスアプリケーションの振舞いの表現をシナリオ記述と呼ぶ。

シナリオ記述のプロセスは、以下の要素から構成される。

- アクティビティ：プロセス中での作業単位
- ポートタイプ：プロセスと FE インタフェースあるいは環境インタフェースとの、データ・制御の入出力を規定
 - FE インタフェース：プロセスと FE とのデータ・制御の入出力を規定
 - 環境インタフェース：プロセスと FE 以外とのデータ・制御の入出力を規定
- リンク：アクティビティ間の制御依存関係
- コンテナ：アクティビティ間で共有されるデータ格納場所

シナリオ記述の問題点として、まずは記述量が膨大になるという点が挙げられる。シナリオ記述は複雑な状態遷移を表現する必要があり、サービスの仕様変更が容易ではない [3]。また、シナリオがパターン化されているにも関わらず、毎回記述しなければいけないという点も問題点である。

2.4 トポロジ記述

ゆかりカーネルにおける FE 間の接続の表現をトポロジ (topology) 記述と呼ぶ。

以下はトポロジ記述の例である。

[FE]

FE_GenerateText, GENERATE, NONE, IMAGE, 1, START, NORMAL, 110

FE_ConsumeText, CONSUME, IMAGE, NONE, 1, START/STOP, NORMAL, 110

[TP]

FE_GenerateText, FE_ConsumeText

左から FE 名, 操作種類, 入力メディア, 出力メディア, 入力ポート数, 制御内容, 優先度, タイムアウト時間を表している。

2.5 モデル変換ツール

すでに既存の研究の一つとしてシナリオの開発支援を目的として設計されたモデル変換ツール基盤が提案されている。

これはトポロジ記述の要素, シナリオ記述の要素のメタモデルを作成し, それに基づいてデータベーススキーマの定義をおこない, 記述要素をデータベース上に格納してモデル変換をおこなう枠組である。

しかし, その問題点として, データベースを用意する必要があり, 複雑な処理により不具合が発生する可能性があることが挙げられる。

次に示すのはモデル変換の手順である。

- (1) 開発者が拡張トポロジ記述を作成し, その作成した拡張トポロジ記述をデータベースに自動挿入

する。

- (2) 指定した通信オペレーションを参照し, 対応するシナリオ記述の構造パターンを選択する。
- (3) 選択したシナリオ記述の構造パターンに, トポロジ記述の要素を自動挿入する。
- (4) トポロジ記述の要素を挿入し終えたデータベースを実行形式に自動変換する。
- (5) シナリオ記述が完成される。

3 サービス記述開発支援ツール

通信パターン, 使用オペレーション, 機能間の接続関係からシナリオ記述を自動生成するツールを設計する。構文解析ルーチン (パーザ) と字句解析ルーチン (スキャナー) を生成するツールとして JavaCC を使用する [4]。JavaCC を使用する理由は, 以下の 2 点が挙げられる。

- データベースを用意する必要がなく, ツールの開発の手間が省ける点
- 変更には弱い, 単純な処理のため不具合が少なく信頼性が確かな点

3.1 生成方法

ユーザが作成した拡張トポロジ記述から通信パターンを参照し, 抽出したテンプレートに機能間の接続関係の情報を挿入する (図 3)。テンプレートは通信パターンごとに用意して, 機能間の接続関係を示している部分だけを拡張トポロジ記述から挿入する。

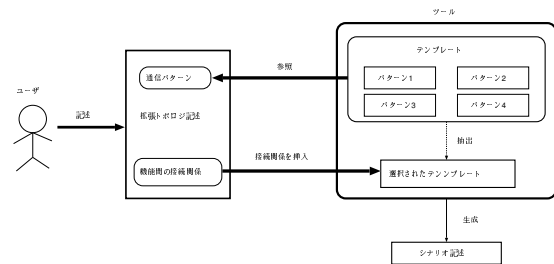


図 3 生成方法

3.2 通信パターン

通信パターンによりシナリオの枠組を決定する (図 4)。

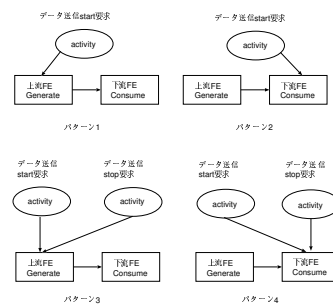


図 4 通信パターン

- パターン 1: 上流 FE から命令 . 下流 FE で止めない .
- パターン 2: 下流 FE から命令 . 下流 FE で止めない .
- パターン 3: 上流 FE から命令 . 下流 FE で止める .
- パターン 4: 下流 FE から命令 . 下流 FE で止める .

3.3 拡張トポロジ記述

現在のトポロジ記述では FE 間のつながりの情報が記述されていない . そこで我々は , ユーザが自動生成ツールによってトポロジ記述からシナリオ記述を変換するために , 拡張トポロジ記述を提案する . 通信パターンによりシナリオの枠組が決定されるので , 拡張トポロジ記述は図 5 のような形になる .

	上流FEの種類	下流FEの種類	上流FE使用オペレーション	下流FE使用オペレーション
パターン1	任意	上流FEの種類による	start.doSend	start
パターン2	任意	上流FEの種類による	start	start.doFetch
パターン3	任意	上流FEの種類による	start.stop.doSend	start.stop
パターン4	任意	上流FEの種類による	start.stop	start.stop.doFetch

図 5 拡張トポロジ記述

3.4 暗黙定義のメッセージタイプとコンテナの関連付け

コンテナの要素を新規にテンプレートに組み込むにあたり , メッセージタイプごとにコンテナ名を決定しておく . また , テンプレートに新しいアクティビティを組み込む場合 , そのアクティビティで使用されているオペレーションからメッセージタイプを判断し , コンテナを自動生成する . 図 6 は使用オペレーション , 加えて暗黙定義されている変数名から適切なメッセージタイプを判断し , 各アクティビティの属性で使用する方法を表している .

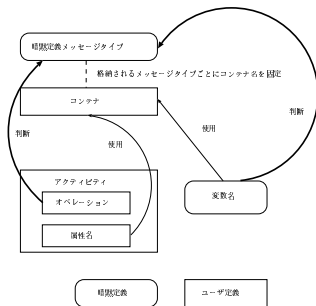


図 6 暗黙定義のメッセージタイプからの自動生成

3.5 シナリオの同期

拡張トポロジ記述にリンクの情報を付加することにより , シナリオ記述の同期を実現する . 洗濯機の洗濯終了の合図を , 文字を表示するだけでなく音も同時に鳴らす場合に , 2 つの FE の接続をどこかで同期させなければ

いけない . 図 7 は , 点線がシナリオ同士のリンクを表している . 図 8 と図 9 はリンクの組み合わせを表している . 「」は片方のシナリオのリンクがもう片方のシナリオの処理が行われるための制約になっていることを表しており , 「x」は制約になっていないことを表している . この表のリンクのパターンを拡張トポロジに付加することで 2 対 2 のシナリオを生成する . 特に 2 つのシナリオを同期させる場合は , 事前条件と事後条件がそれぞれパターン 1 になる .

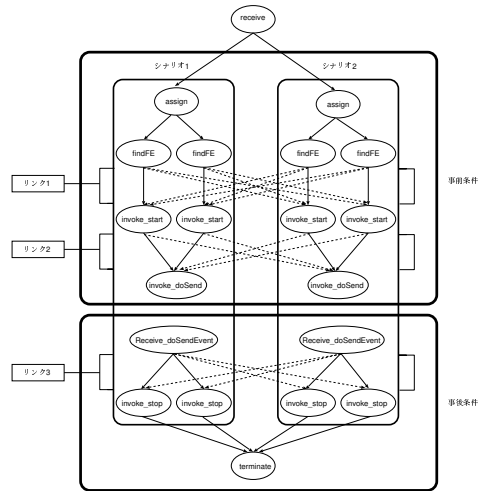


図 7 シナリオの同期

	シナリオ1		シナリオ2	
	リンク1	リンク2	リンク1	リンク2
パターン1	○	○	○	○
パターン2	○	○	○	x
パターン3	○	○	x	○
パターン4	○	○	x	x
パターン5	○	x	○	○
パターン6	○	x	○	x
パターン7	○	x	x	○
パターン8	○	x	x	x
パターン9	x	○	○	○
パターン10	x	○	○	x
パターン11	x	○	x	○
パターン12	x	○	x	x
パターン13	x	x	○	○
パターン14	x	x	○	x
パターン15	x	x	x	○
パターン16	x	x	x	x

図 8 同期箇所の組み合わせ (事前条件)

	シナリオ1	シナリオ2
	同期箇所3	同期箇所3
パターン1	○	○
パターン2	○	x
パターン3	x	○
パターン4	x	x

図 9 同期箇所の組み合わせ (事後条件)

4 評価と考察

4.1 データベース化についての考察

サービス開発支援の一つの方法として、プロセスの要素をコンポーネントとして、それをデータベースとし、開発者がシナリオを記述する際にライブラリからコンポーネントを取り出し、コンポーネントからシナリオを構築する方法を考察する。データベースにすることによって、ユーザがコンポーネントを組み合わせるだけでシナリオを記述することができる。

プロセスの要素をコンポーネントとし、それをデータベースにする方法は、それをライブラリから取り出すだけでシナリオ記述を書くことができる。しかし、アクティビティとオペレーションの組合せにより、コンポーネントが膨大な量になってしまう。図 10 はプロセスの各要素をコンポーネントとし、それをデータベースとしている。データベースに格納した後、データベースからコンポーネントを取り出して、コンポーネントからシナリオを構築する。以下はコンポーネントの種類である。

- コンテナコンポーネント:使用するメッセージタイプごとにコンテナ名を固定して、コンテナの集合をコンポーネントとする。
- パートナーリンクコンポーネント:FE の接続の数だけコンポーネントが存在する。
- リンクコンポーネント:リンクの集合をコンポーネントとする。各アクティビティの適当な名前をつけ、先に定義するような形とする。
- アクティビティコンポーネント:各アクティビティにて、使用されるオペレーションの数だけコンポーネントが存在する。

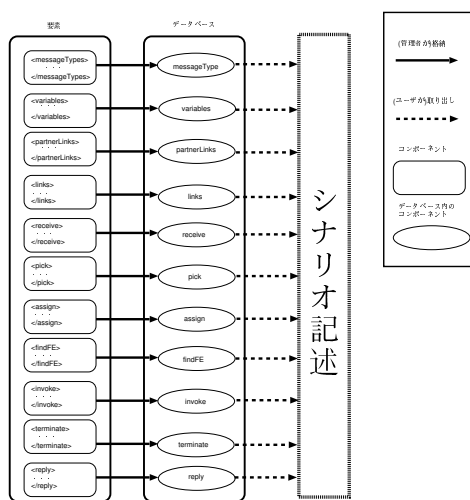


図 10 コンポーネントのデータベース

4.2 機能間の接続が多対多の場合のシナリオ記述の自動生成の考察

FE の接続が多対多の場合も、リンクのタイミングを拡張トポロジ記述に付加することで自動生成できると考えている。その場合、リンクのパターンが大幅に多くなる。特に、シナリオを同期させる場合にどの部分をどの部分をリンクされるかを考える必要がある。

5 おわりに

本研究では、まず FE の 1 対 1 接続の場合の拡張トポロジ記述を定義した。シナリオ記述の生成方法として以下の方法を提案した。

- (1) 拡張トポロジ記述から通信パターンを参照
- (2) その通信パターンに対応したテンプレートが選択される
- (3) 選択されたテンプレートに機能間の接続の情報を挿入し、シナリオ記述を生成する

また、コンテナの要素を新規にテンプレートに組み込むにあたり、暗黙定義のメッセージタイプからコンテナを自動生成し、あらかじめ用意してあるテンプレートに当てはめる方法を提案した。また、FE の 1 対 1 接続の場合の拡張トポロジ記述を定義した。さらに、拡張トポロジ記述に付加するリンク箇所について考えた。以下の 2 点を今後の課題とする。

- コンポーネントのデータベース化
- FE が多対多の接続の場合のシナリオ記述の自動生成

コンポーネントをデータベースとするにあたり、データベースの開発言語を選択する必要がある。データベースにする問題点として、コンポーネントの量が膨大になる点が挙げられるので、それを改善する方法を考案する必要がある。FE が多対多の接続の場合、特にシナリオを同期させる場合のリンクのパターンを考える必要がある。

参考文献

- [1] 沢田 篤史, 多鹿 陽介, 山崎 達也, 美濃 導彦 : “ゆかりコア: ネットワーク家電のための分散協調型サービス構築基盤”, 電子情報通信学会技術研究報告, ソフトウェアサイエンス研究会 (SS2004-9), vol.104, pp.19-24, 2004.
- [2] 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.
- [3] 坪井正徳, 沢田 篤史, 河原達也 : “モデル駆動アーキテクチャに基づくホームネットワークサービスの開発支援”, 情報処理学会研究報告 (NO.2007-SE-155), pp.1-8, 2007.
- [4] 五月女健治 : “JavaCC-コンパイラ・コンパイラ for Java”, テクノプレス (2003).