社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

センサ駆動連携サービスのための連鎖検出手法の検討

稲田　卓也† 　池上　弘祐† 　まつ本真佑† 　中村　匡秀† 　井垣　宏††

† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1–1
†† 東京工科大学　〒 192–0982 東京都八王子市片倉町 1404-1

あらまし　家庭内のネットワーク（ホームネットワーク）を利用し，複数のセンサおよび家電機器を連携する家電連携サービスの研究・開発が進んでいる本稿では，センサと家電機器が連携する家電連携サービスを**センサ駆動連携サービス** (センササービス) と定義し，連携サービス間に発生する**連鎖**を検出するシステムを提案する．連携サービス間の連鎖とはある連携サービスの実行結果が与えた環境の変動をセンサ駆動連携サービスがイベントと検知してしまうことを指す．このような連携サービス間連鎖は，開発者の意図しない不具合を引き起こす可能性がある．我々は先行研究で提案した連鎖検出アルゴリズムにもとづき，サービス間連鎖検出システムを開発した．さらに，14 人の被験者を対象として 7 種類のサービスから連鎖を検出してもらい，我々のシステムの結果と比較する実験を行った．連鎖検出実験により，人手で正確に連鎖を検出することが困難であることと我々の連鎖検出システムが有用であることが確認できた．

キーワード　HNS, 家電連携サービス, センサ駆動, サービス競合，サービス連鎖

A Study of Detecting Chain Reaction between Sensor Driven Services

Takuya INADA†, Kosuke IKEGAMI†, Shinsuke MATSUMOTO†, Masahide NAKAMURA†, and Hiroshi IGAKI††

† Kobe University　Rokkoudaityou 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
†† Tokyo University of Technology

Abstract　Appliance integration services which make two or more sensors and home appliances in home network system are widely developed. In this paper, we define an appliance integration service which cooperate sensors and appliances as a sensor-driven service, and propose a chain reaction detection system between an appliance integration service and a sensor-driven service. The chain reaction between an appliance integration service and a sensor-driven service occurs by detecting execution result of the integration service as an event for the sensor-driven service. Such chain reactions may cause unintended problems for developers and users. We have proposed an algorithm to detect such chain reactions in our previous study. In this study, we developed a chain reaction detection system based on the algorithm. In addition, we performed experiments to evaluate difficulty of detecting chain reactions between integration services manually.

Key words　HNS, Integration Services, Sensor-Driven Service，Feature Interaction，Chain Reaction

1. はじめに

複数の家電機器やセンサを宅内ネットワーク (ホームネットワーク) に接続し，連携制御する**ホームネットワークシステム (HNS)** の普及・導入が進んでいる [1] [2]．HNS では，テレビや DVD レコーダ，エアコン，照明，扇風機といった複数の家電や人感センサ，照度センサといったセンサを連携制御することで，ユーザによる家電の単体制御と比べてより付加価値の高い**家電機器連携サービス** (以下，連携サービス) を実現することができる．また，連携サービスの中でも特にセンサによるイベント検知を契機として動作する**センサ駆動連携サービス** (センササービス) は，照明機器と人感センサ，自動ドアやセキュリティアラーム等実際に数多くの製品が開発・販売されている [3] [4] [5]

以下にセンサ駆動連携サービスの例を紹介する．

SS1：自動照明点灯サービス　:暗くなったら，調光機能を使って LivingLight1 の明るさを 10 にセットする．

SS2：長期不在省エネサービス　:部屋に人が居なくなってか

ら 30 分が経過したら，TV1, DVDPlayer1, AirConditioner1, LivingLight1 の電源を落とす．

SS1 では，部屋の明るさを検知する照度センサが，SS2 では人の入退室状態を検知する入退室センサが利用されている．各センサは室温，照度といった部屋の環境状態の変化をイベントとして検知する．検知内容が連携サービスで指定されたものであれば，サービス内で定義された処理が実行される．

連携サービスには，照明機器，空調機器のような，機器の実行結果が環境に対して影響をあたえるものが存在する．そのため，センサ駆動連携サービスではある連携サービスの実行結果が他の連携サービスの開始/終了に影響を与えることによる不具合を考慮する必要がある．例えば上記の連携サービス SS2 をユーザが実行することで部屋の明るさは暗くなる．SS2 は照度センサによって部屋の暗さをイベントとして検知するため，SS1 が実行されることで連鎖的に SS2 が実行されて部屋が明るくなってしまうことがある．

我々は先行研究において，このような連携サービスの実行結果が他のサービスのイベントに波及することを**サービスの連鎖**と呼び，その検出アルゴリズムを提案した．本研究では，我々の提案した検出アルゴリズムにもとづくサービス連鎖検出システムの開発を行い，7 種類の連携サービスを対象に連鎖検出実験を行った．連鎖検出実験では，実際に 14 名の被験者を対象として連鎖元サービス，連鎖先サービス，連鎖が発生する環境プロパティ，連鎖が発生する条件，連鎖を引き起こす処理の 5 項目をの連鎖がどの程度正確に検出できるかを評価した．その結果，連鎖元サービス，連鎖先サービス，連鎖が発生する環境プロパティについてはほとんどの被験者が正確に解答できたが，連鎖が発生する条件，連鎖を引き起こす処理の部分については不正確な解答が非常に多くなった．すなわち，我々の提案するサービス連鎖検出システムが連携サービスおよびホームネットワークシステム開発において非常に有用であることが確認できた．

2. 準　　備

2.1 ホームネットワークシステム (HNS)

ホームネットワークシステム (HNS) は，宅内のネットワークに接続された複数のネットワーク家電から構成されている．各家電機器は，ユーザや外部エージェントがネットワーク越しに制御できるように，機器メソッドと呼ばれる制御 API を備えている．機器メソッドを呼び出すことによって，各家電機器の内部状態 (機器状態) が変更され，結果として機器の振る舞いが変更される．我々は機器メソッドと機器プロパティから構成される機器のモデルを先行研究 [6] において機器オブジェクトモデルと定義している．機器メソッドを組み合わせることで，宅外からの制御や連携サービスといった各種アプリケーションが実際に開発・提案されている．

我々の研究室では，実際の HNS(CS27-HNS と呼ぶ) [8] を構築しており，エアコン，照明，TV といった多様な家電の機能が Web サービス [13] として公開されている．Web サービスとして構築された各機器の API は SOAP または REST 形式で呼び出すことが可能であり，複雑な連携サービスでも容易に開発することが可能となっている．

2.2 センサ駆動連携サービス

ホームネットワークシステム内において，センサによって人の出入りや室温などの環境の変化をイベントとして検知し，イベントに応じた振る舞いを提供するサービスをここでは**センサ駆動連携サービス** (センササービス) と呼ぶ．例えば,「室温が 28 ℃以上になる」というイベントを温度センサが検知し,「エアコンの冷房運転を開始する」といったような機器連携が行われる．

以下に本稿で対象とするセンササービスの例をあげる．

SS1：自動照明点灯サービス　:暗くなったら（明るさが 200lx 未満），調光機能を使って LivingLight1 の明るさを 10 にセットする (setBrightness(10))．

SS2：長期不在省エネサービス　:部屋に人が居なくなってから 30 分が経過したら，TV1, DVDPlayer1, AirConditioner1, LivingLight1 の電源を落とす (全ての機器について off())．

SS3：タイマー空調 ON サービス　:タイマー空調セットボタンが ON であるとき，設定された開始時刻 (7:00) に AirConditioner1 の暖房が 26 度設定で実行される (heating(26))．その後設定された終了時刻 (9:00) に AirConditioner1 は OFF になる．※なお，終了時刻までには 26 ℃に達するとする

SS4：電力連動省エネサービス　:消費電力が 1500W を 5 分間超え続けた場合に AirConditioner1 もしくは LivingLight1 の状態が OFF でなければ，AirConditioner1 の設定温度を 20 度にし (heating(20))，LivingLight1 の調光を 2 段階暗くする (downLightLevel(2))．

SS5：お出かけサービス　:おでかけボタンを押すと，TV1, DVDPlayer1，AirConditioner1，EntranceLight1 の off() が実行され，Curtain1 が閉じられる (close())．

SS6：自動暖房サービス　:部屋に人がいるときに室温が 14 度未満になったら，AirConditioner1 の暖房を 26 度設定でつける (heating(26))．

SS7：自動冷房サービス　:部屋に人がいるときに室温が 25 度より暑くなったら，自動で AirConditioner1 の冷房を 24 度設定でつける (cooling(24))．

これらのシナリオでは家電機器の処理説明に続けて，その機器のどの機器メソッドが実行されるかが括弧内で記述されている．

我々はセンサ駆動サービスを ECA ルール [9] にもとづいて記述する．ECA はそれぞれ Event，Condition，Action を指している．Event はサービスを実行する契機となるトリガやきっかけを表す．Condition はイベントが発生した際に評価される論理式であり，これが True であるときにのみサービスの振る舞いである Action が実行される．我々は先行研究 [12] において，ECA ルールにもとづくセンササービスの記述方法を提案した．我々が提案したセンササービス記述は ***S_event***，***S_condition***，***S_action*** の 3 つの要素を持つ．それぞれの詳細は以下に示すとおりである．

- ***S_event*** :センササービスを実行する契機となるイベン

トを機器や環境のプロパティから構成される条件式として記述する．ここで環境プロパティはセンサが取得する環境の状態を示す [12]．環境プロパティは env.Brightness が室内の照度を env.Temperature が室温といったように，”env.”ではじまる形式で記述する．”env.”が共通の環境プロパティは同じ空間のプロパティであることを示している．例えば,「室温が 28 ℃を超える」というイベントをトリガにしてサービスが起動される場合は，$env.Temperature > 28(℃)$ と記述される．

- ***S_condition*** :*S_event* が検知された際に，*S_action* の実行可否を判断するための条件式として記述する．条件式の構成要素は *S_event* と等しい．例えば,「室温が 28 ℃を超える」というイベントをトリガにして起動されるサービスが,「室内に人が居たら」という条件が成立するときにのみ *S_action* が実行されるとする．この場合は *S_condition* として，$env.PeopleNum > 0$ が定義される．
- ***S_action*** :連携サービスシナリオとして，実行される機器メソッドの系列を記述する．

S_action の持つ各機器メソッドは以下の ***ME_future*** を持つ．

- ***ME_future*** :ある状態にある機器の機器メソッドを実行し続けたときに特定の環境プロパティに与える最終的な影響の大きさと方向を示す．我々の提案では，機器メソッドによる機器状態の変化を状態遷移機械を用いたモデルとして表す．

センササービス SS1 SS7 のセンササービスのサービス記述を図 1 に示す．また，これらのサービスに含まれる AirConditioner，TV，Light といった機器の ME_future を表 1〜3 に示す．

2.3 センサ駆動連携サービスにおけるサービス競合

センササービスでは，室温や湿度，明るさの変動といった環境の変化をセンサによってイベントとして検知し，対応する機器制御などを行う．そのため，あるサービスの機器制御を実行した結果が他のサービスのイベントを誘発することがある．先行研究 [12] ではこのような連携サービスにおけるイベントの誘発を**サービスの連鎖**と呼ぶ．連鎖によってユーザの意図しないサービスが実行されてしまうことがある．前節であげた連携サービスを例に，連鎖が発生する 2 つのシナリオを紹介する．

シナリオ C1 : SS3:タイマー空調 ON サービスによって，暖房が 26 度設定で実行され，部屋の室温が 26 度になると，SS7:自動冷房サービスの実行条件である「部屋に人がいるときに室温が 25 度より暑くなったら」を満たしてしまう可能性がある．すなわち，SS3 の実行結果が SS7 の実行を誘発する連鎖が発生する．

C1 のケースでは，通常 SS7 は暖房が実行されるような季節に実行されるべきではないと考えられる．しかしながら，この事例のような連鎖が実際に発生するということが事前に分かっていなければ，ホームネットワークシステムを実行する際の障害となる可能性がある．

このように，センササービスは，他の連携サービス実行の影響を受けることが多い．また，事前にどのような連鎖が発生する可能性があるかは十分に調査されている必要がある．我々は連鎖によって連携サービスの実行結果がユーザの意図しないものになってしまうことを**センササービスにおけるサービス競合**と呼ぶ．

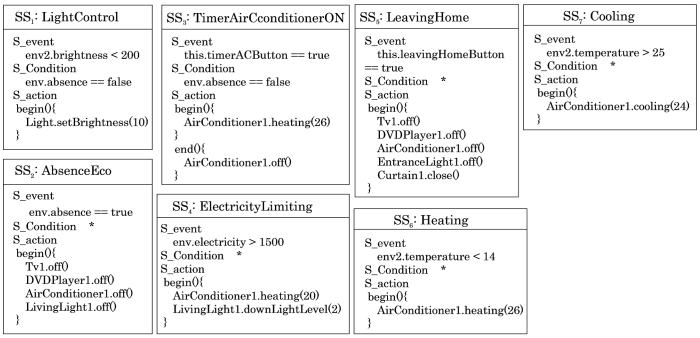

図 1　センサ駆動連携サービスシナリオ

表 1　AirConditioner ME_future

State \ Method	off()	heating(heatTemp)	cooling(coolTemp)
OFF	next : OFF	env.electricity += 1500 env2.temperature = heatTemp next : HEATING	env.electricity += 1500 env2.temperature = coolTemp next : COOLING
HEATING	env.electricity −= 1500 env2.temperature = 10 next : OFF	next : HEATING	env2.temperature = coolTemp next : COOLING
COOLING	env.electricity −= 1500 env2.temperature = 10 next : OFF	env2.temperature = heatTemp next : HEATING	next : COOLING

表 2　TV ME_future

State \ Method	off()	on()
OFF	next : OFF	env2.sound += 10*volumeLevel electricity += 500 next : ON
ON	env2.sound −= 10*volumeLevel electricity −= 500 next : OFF	next : ON

表 3　LivingLight ME_future

State \ Method	off()	on()	setBrightness(setB)
OFF	next : OFF	env2.brightness += 100*brightnessLevel env.electricity += 50 next : ON	env2.brightness += 100*setB env.electricity += 50 brightnessLevel = setB next : ON
ON	env2.brightness −= 100*brightnessLevel env.electricity −= 50 next : OFF	next : ON	env2.brightness += 100*(setB−brightnessLevel) brightnessLevel = setB next : ON

次節では，このようなセンササービスにおけるサービス競合検出を目的とした連携サービス間の連鎖を検出する手法についての我々の先行研究 [12] を紹介する．

2.4 連鎖検出手法

我々が先行研究において提案したセンササービス間の連鎖検出手法を示す．ここでは先に実行される連携サービスを SS_{old}，また SS_{old} 実行の結果，連鎖が引き起こされる可能性のある連携サービスを SS_{new} とし，2 者間に連鎖の関係があるかどうか，またどのような機器状態および環境状態のときに連鎖が発生するかを検出する手法を示す．

検出は以下の手順で行う．

Step1 : SS_{new} からイベント SS_{new_event} を取得する．

Step2 : SS_{new_event} に含まれている全てのプロパティの集合 P を抽出する．

Step3 : SS_{old} の SS_{old_action} に含まれる全ての機器メソッドのうち，P に含まれるプロパティを一つ以上含む ME_future

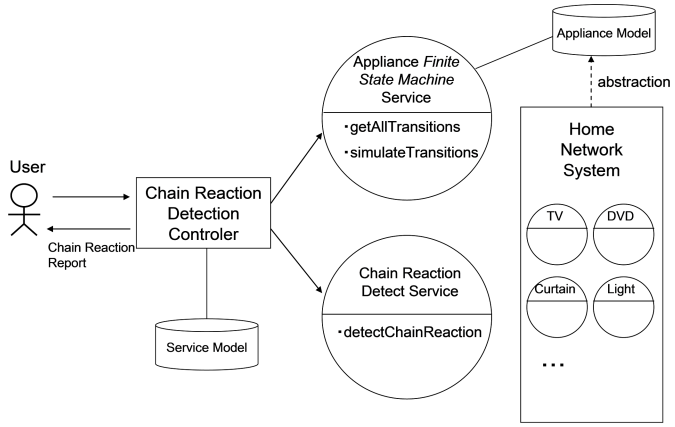

図 2　サービス連鎖検出システムアーキテクチャ図

を持つ機器メソッド集合 $M_{P_{SS_{old}}}$ を抽出する．

Step4： $M_{P_{SS_{old}}}$ に含まれる機器メソッド $m1$, $m2$,.., mn それぞれがもつ ME_future 集合 (F_{m1},F_{m2} ,... ,F_{mn}) からプロパティ P を含むもの $F_{P_{mi}}(1 \leqq i \leqq n)$ をメソッド毎に抽出する．

Step5： $F_{P_{mi}}(1 \leqq i \leqq n)$ における全ての ME_future の組み合わせ集合 FC_{ME_future} を算出する．

Step6： 全ての $fc_{ME\_future}(\in FC_{ME\_future})$ において，プロパティ$p(\in P)$ 毎の ME_future を加算した合成 ME_future, $afc_{ME\_future}(\in AFC_{ME\_future})$ を算出する (加算方法については後で詳述する)．

Step7： 全てのプロパティ$p(\in P)$ の初期状態が SS_{new_event} の補集合に含まれているという仮定の下で，SS_{new_event} の補集合から SS_{new_event} の状態に遷移させる可能性がある全ての afc_{ME_future} を AFC_{ME_future} から導出する．ここで afc_{ME_future} が 1 つでも導出されれば，SS_{new} は SS_{old} に連鎖する．

Step8： Step7 で導出された afc_{ME_future} をもとに，連鎖を起こす可能性がある初期の環境状態および機器状態を導出する．

以上のような連鎖検出プロセスによって，連携サービス間の連鎖と連鎖が発生する状況を検出することが可能となった．本研究では，我々が提案した連鎖検出プロセスにもとづいて連鎖の検出を実際に行うサービス連鎖検出システムを開発し，その有用性を評価するための連鎖検出実験を行った．次節では，我々が開発したサービス連鎖検出システムについて詳述する．

3.　サービス連鎖検出システム

3.1　アーキテクチャ

図 2 に我々が開発したサービス連鎖検出システムのアーキテクチャを示す．システムは，**連鎖検出コントローラ**，**状態遷移機械サービス**，**連鎖検出サービス**の大きく 3 つのコンポーネントから構成される．連鎖検出コントローラはユーザからの要求を受け，連鎖検出結果を返す．連鎖検出は連鎖検出コントローラを通じて状態遷移機械サービスと連鎖検出サービスを利用することで行われる．以下の節で各構成要素の説明を行う．

3.2　連鎖検出コントローラ

コントローラは，ユーザからのサービス連鎖検出要求を受けつけ，状態遷移機械サービス，連鎖検出サービスを利用し連鎖検出を行う．このコントローラでは 2.4 節の Step1〜3，5 が実現される．

ユーザが連鎖検出を実行すると，連携サービスのシナリオが定義された外部 XML ファイルより，S_event，S_condition，S_action を読み込みサービスオブジェクトを生成する (Step1,2)．さらに，S_action に含まれるメソッド毎に状態遷移機械サービスから ME_future を取得し，メソッド下にリストで保持する (Step3)．そして，サービスが持つメソッド毎の ME_future を組み合わせ FC_{ME_future} を生成する (Step5)．生成された全ての $fc_{ME\_future}(\in FC_{ME\_future})$ を再び状態遷移機械サービスに渡し，返り値として $afc_{ME\_future}(\in AFC_{ME\_future})$ を受け取る．サービスオブジェクトはこれらの afc_{ME_future} をリストで保持する．

全てのサービスについて afc_{ME_future} を獲得した後，連鎖検出サービスに連鎖元サービスの afc_{ME_future} と連鎖先サービス S_event をセットで渡す．連鎖検出サービスは実際に連鎖検出を行い，連鎖を引き起こす可能性がある環境プロパティ毎の初期状態を出力する．これらの作業をサービスの全組み合わせについて行う．連鎖検出サービスの実行結果にもとづき，コントローラが結果出力を行う．出力されるファイルには，連鎖元サービス，連鎖先サービス，連鎖 ID，連鎖が発生する環境プロパティ，連鎖が発生する条件，連鎖を引き起こす処理の情報が含まれる．

3.3　状態遷移機械サービス

状態遷移機械サービスは，機器毎の ME_future を保持する．連鎖検出の際，2.4 節の Step4，6 にあたる作業を行う．このサービスは，ME_future の定義を含む機器を定義した外部 XML を読み込み，機器毎の状態遷移機械オブジェクト AFSM を生成・保持する．

このサービスは以下のメソッドを外部に公開している．

getAllTransitions(String applianceName, String methodName):Transition[] 引数が機器名およびメソッド名，戻り値は Transition 型の配列となっている．このメソッドは，引数として与えられた機器のメソッドが持つすべての ME_future を配列にして返す (Step4)．Transition クラスは ME_future をクラス化したものであり，属性は以下のとおりである．

Transition
- method(String)：機器メソッド名.
- params(String[])：機器メソッドの引数.
- preState(String)：遷移前状態.
- nextState(String)：遷移後状態.
- effects(Effect[])：環境への影響値の配列.

simulateTransitions(Transition[] transitions) :EProperty[] Transition クラスの配列を引数として受け取り，EProperty クラスの配列を返す．与えられた Transition 配列の環境への影響値をすべて計算し，影響値を EProperty の配列

として返す (Step6)．は環境への影響値を表す．また返り値の EProperty の属性は以下のとおりである．

EProperty

- ePropertyName(String)：環境プロパティ名．
- formula(String)：環境への影響を表す数式．
- numericable(boolean)：true なら数値，false なら文字列を表す．
- cumulation(boolean)：true なら加算可能，false なら加算不可能．

3.4 連鎖検出サービス

連鎖検出サービスは，連鎖元サービスの環境への影響値と連鎖先サービスのイベントから連鎖を起こす可能性がある初期の環境状態を導出する．2.4 節の Step7，8 にあたる．

このサービスは以下のメソッドを外部に公開している．

checkChainReaction(EProperty[] ePropertys, String event):EPropertyRange[] 連鎖元サービスの影響値 (EProperty 配列) とイベント (String 型) を引数とし，連鎖を起こす可能性がある初期の環境状態を導出する (Step7，8)．返り値の EPropertyRange クラスは個別の環境プロパティの数値範囲を表す．

4. 実　　験

4.1 実験概要

サービス間の連鎖を実際に手作業で検出してもらい，連鎖検出がどの程度困難であるかを評価する実験を行った．被験者には，2. 章の 7 つのサービスシナリオ (そのサービスで実行される機器機能を示したもの) とサービスで利用される機器の状態遷移表 (ME_future) を与える．それらをもとに，下記のフォーマットにしたがって連鎖を記述してもらう．実験時間は 2 時間，被験者は 14 名を対象に行った．

連鎖元サービス　:連鎖を起こす元となるサービス名．

連鎖先サービス　:連鎖元サービスの実行がきっかけで連動して動作するサービス名．

連鎖が発生する環境プロパティ　:連鎖の原因となる環境プロパティ名．

連鎖が発生する条件　:連鎖が発生する環境プロパティやセンサが示す値の範囲を条件式として記述したもの．

連鎖を引き起こす処理　:連鎖元サービス内の連鎖を引き起こす具体的な処理 (機器機能)．

図 3 は，実験で想定する部屋の環境である．部屋は玄関とリビングの 2 つが存在し，それぞれ env1, env2 で始まる環境プロパティを持つ．具体的には玄関は env1.brightness (玄関の明るさ．単位：lx)，リビングは env2.brightness(リビングの明るさ．単位：lx) および env2.temperature(リビングの室温．単位：℃) といった環境プロパティを保持している．さらに，これら 2 つの部屋に共通の環境プロパティとして env.electricity(全体の消費電力．単位：W) がある．使用する機器は，玄関に玄関照明，リビングに TV，DVDPlayer，リビング照明，エアコン，床暖房，カーテン，セキュリティカメラ．部屋に配置された各家電機器が環境プロパティに与える影響は，機器ごとの状態遷移表によって示され，2. 章の表 1〜3 をすべての機器について与えた．

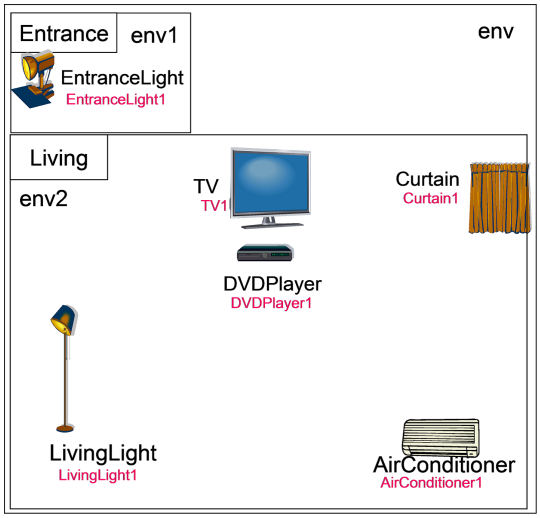

図 3　実験環境図

4.2 実験結果

表 4 に実験結果を示す．図はそれぞれ解答フォーマットの項目ごとの正解数，精度，再現率を表している．サービスレベルは，連鎖元サービスと連鎖先サービスが模範解答と一致している場合を正解としてカウントした．プロパティレベルは連鎖が発生する環境プロパティ，処理レベルは連鎖を引き起こす処理，条件レベルは連鎖が発生する条件を正解単位とした．

その結果，サービスレベルおよびプロパティレベルでは精度，再現率ともに平均 80%以上であったが，処理レベルでは精度 60%, 再現率 51% になり，条件レベルでは精度 36%，再現率 30 %まで悪化した．

処理レベルにおける不正解事例では，長期不在省エネサービスと自動照明点灯間の連鎖の [連鎖を引き起こす処理] 項目で，「機器状態が ON(brightnessLevel = 1) である LivingLight1 において off() を実行」が正解のところ，「機器状態が ON である LivingLight1 において off() を実行」というように，機器プロパティまで考慮にいれて場合分けしていない事例が目立った．この場合では，[連鎖が発生する条件] 項目が，正解が「200 <= env2.brightness < 300」のところ，「200 <= env2.brightness < 1200」というように，機器プロパティまで考慮にいれて場合分けしていないので，範囲が大きく [連鎖が発生する条件] 項目でも不正解であった．

また，お出かけサービスと自動暖房サービス間の連鎖で [連鎖が発生する条件] 項目の正解が，「14 <= env2.temperature and 在室人数 > 0」のところ「14 < env2.temperature and 在室人数 >= 1」というように，env2.temperature に関する条件の符号に等号が含まれておらず間違っている．このように，環境プロパティに関して，符号に関する間違いも目立った．他の事例としては自動照明点灯サービスと電力連動省エネサービス間連鎖に関して，「1450 < env.electricity <= 1500」が正解のところ「env.electricity <= 1500」のように，下限が無い間違いがあった．

表 4 実験結果

	サービスレベル			プロパティレベル			処理レベル			条件レベル		
名前	正解数	精度 (%)	再現率 (%)	正解数	精度 (%)	再現率 (%)	正解数	精度 (%)	再現率 (%)	正解数	精度 (%)	再現率 (%)
A	10	100	83.3333	28	100	75.6757	18	64.2857	48.6486	3	10.7143	8.10811
B	11	84.6154	91.6667	36	100	97.2973	11	30.5556	29.7297	14	38.8889	37.8378
C	9	100	75	23	100	62.1622	3	13.0435	8.10811	0	0	0
D	12	100	100	37	100	100	27	72.973	72.973	13	35.1351	35.1351
E	12	85.7143	100	37	100	100	23	62.1622	62.1622	14	37.8378	37.8378
F	4	50	33.3333	6	100	16.2162	3	50	8.10811	3	50	8.10811
G	7	50	58.3333	23	100	62.1622	6	26.087	16.2162	0	0	0
H	10	100	83.3333	27	96.4286	72.973	13	46.4286	35.1351	11	39.2857	29.7297
I	11	100	91.6667	35	100	94.5946	34	97.1429	91.8919	31	88.5714	83.7838
J	11	73.3333	91.6667	31	93.9394	83.7838	28	84.8485	75.6757	29	87.8788	78.3784
K	8	100	66.6667	28	100	75.6757	21	75	56.7568	10	35.7143	27.027
L	11	100	91.6667	36	100	97.2973	35	97.2222	94.5946	14	38.8889	37.8378
M	12	100	100	37	100	100	22	59.4595	59.4595	16	43.2432	43.2432
N	11	100	91.6667	35	100	94.5946	24	68.5714	64.8649	1	2.85714	2.7027

連鎖について分析を行う場合，このような現在の機器状態および機器が持つ内部プロパティについて考慮することや環境プロパティの変動内容を正確に把握することは非常に重要である．一方で，本稿で提案するような検出を支援するためのモデルおよびツールが無ければ，正確な連鎖検出が困難である．結果として，我々のサービス連鎖検出システムとサービスモデルが今回の実験で対象としたような複数の部屋および家電機器から構成される連携サービス間の連鎖検出において，非常に有用であることが確認できた．

5. おわりに

本稿では，先行研究に置いて提案したセンササービスを含む連携サービス間に発生するサービスの連鎖検出アルゴリズムにもとづいて，サービス連鎖検出システムを開発した．さらに，実際に複数の部屋および複数の家電機器から構成されるホームネットワークシステムおよび連携サービスを想定し，連鎖検出実験を行った．連鎖検出実験により，開発時に必要とされる連鎖が発生する条件などの詳細な分析が開発者には困難であることと我々の連鎖検出システムの有用性が明らかになった．

今後はより柔軟性の高い連鎖や検出された連鎖の危険度などを考慮した連鎖検出システムの提案へとつなげていくことを考えている．

謝辞 この研究の一部は，科学技術研究費 (若手研究 B 20700027,21700077)，およびパナソニック電工株式会社の助成を受けて行われている．

文　献

[1] パナソニック電工株式会社,“ ライフィニティ ”, http://denko.panasonic.biz/Ebox/kahs/
[2] 東芝,“ 東芝ネットワーク家電 Feminity ”, http://www3.toshiba.co.jp/feminity/about/index.html
[3] Matsushita Electric Works Ltd. Katteni switch, http://biz.national.jp/Ebox/katte_sw/.
[4] Nabtesco Corp. Automatic entrance system. http://nabco.nabtesco.com/english/door_index.asp.
[5] Daikin Industries Ltd. Air conditioner, http://www.daikin.com/global_ac/.
[6] M. Nakamura, H. Igaki, and K. Matsumoto, ”Feature Interactions in Integrated Services of Networked Home Appliances -An Object-Oriented Approach-,” In Proc. of Int’l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI’05), pp.236-251,2005.
[7] M. Wilson, M. Kolberg, and E. H. Magill. Considering side effects in service interactions in home automation - an online approach. In *Proc. Int’l. Conf. on Feature Interactions in Software and Communication Systems (ICFI’07)*, pages 172–187, 2007.
[8] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. ichi Matsumoto. Constructing home network systems and integrated services using legacy home appliances and web services. *International Journal of Web Services Research*, 5(1):82–98, January 2008.
[9] Chongqing Zhang, Minglu Li and Qunhua Pan, ”An ECA Rules Based Middleware Architecture for Wireless Sensor Networks ”, IEEE, ISBN:0-7695-2405-2 , PDCAT’05, Pages: 586 - 588, 05-December 08, 電子情報通信学会技術研究報告,Vol.108, No.136, pp.35-40,July 2008. 電子情報通信学会技術研究報告, Vol.108, No.458, pp.439-444, March 2009.
[10] 吉村 悠平, 稲田 卓也, 井垣 宏, 中村 匡秀, “ SOA にもとづくホームネットワーク サービス競合検出・解消基盤の提案, ”情報処理学会研究報告, vol.2009-SE-166, no.4, November 2009.
[11] Masahide Nakamura, Hiroshi Igaki, Yuhei Yoshimura, and Kousuke Ikegami, ”Considering Online Feature Interaction Detection and Resolution for Integrated Services in Home Network System, ”In 10th International Conference on Feature Interactions in Telecommunications and Software Systems (ICFI2009), pp.191-206, June 2009.
[12] 稲田卓也, 池上弘祐, まつ本真佑, 中村匡秀, 井垣宏, ”センサ駆動連携サービスのためのサービス競合検出手法に関する検討, ”, no. 信学技報, vol. 110, no. 172, AI2010-15, 電子情報通信学会, August 2010.
[13] web サービス, http://www.w3.org/2002/ws;

付　録

1.