

Title	ホームネットワーク上の家電機器とセンサを用いた双方向通知応答基盤の研究
Author(s)	丸山, 宏介
Citation	
Issue Date	2019-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/15958
Rights	
Description	Supervisor:丹 康雄, 情報科学研究科, 修士 (情報科学)

修士論文

ホームネットワーク上の家電機器と センサを用いた双方向通知応答基盤の研究

1510051 丸山 宏介

主指導教員 丹 康雄 教授
審査委員主査 丹 康雄 教授
審査委員 篠田 陽一 教授
審査委員 リム 勇仁 准教授

北陸先端科学技術大学院大学

情報科学研究科

平成 31 年 2 月

概要

メッセージやアラートにより情報を提示する「通知」というインタフェースについて研究が行われている。ただ、通知の研究はデスクトップPCやスマートフォンを対象としたものがほとんどで、我々の身に寄り添う家電機器からの通知を管理する研究は少ない。家電機器による通知についての研究として、家電機器からユーザに向けて一方的に情報を発信するプッシュ型情報提示システムがある。しかし、プッシュ型情報では、見逃す・聞き逃すことがあってもそれを管理システムやアプリケーションが認知できないという問題点がある。本研究では住環境とユーザの自然な双方向性の情報のやり取りができることを目的とし、これを達成するために家電製品の通知とセンサによるユーザ応答検知ができる双方向通知応答基盤を提案・実装する。まず、ECHONET Lite 機器・oneM2M 家電機器のクラスより、音声・視覚的シグナルを発する 17 機器を調査し、通知について {1. 現状認識, 2. 動作・状態変化, 3. 動作終了, 4. 呼び出し, 5a. 許容値外 a, 5b. 許容値外 b, 6. 異常, 7. タスク, 8. 緊急} の 9 種類に分類した。また、応答について {A. 暗黙応答, B. 明示応答, C. 手続き応答} の 3 つの大区分と {B1. 現場応答, C1. 復帰応答, C2. サービス依存応答} の 3 つの小区分に分類した。双方向通知応答基盤においては、獲得、モデリング、理論・推論、配信の 4 要素の考え方を取り入れ、ユーザを取り巻く環境の情報処理を行う。モデリング部について、コンピュータも開発者も認識しやすい形式のオントロジ言語 OWL による通知応答のコンテキスト記述を行い、RDF を用いてモデルに沿った実データを管理できるようにする。コンテキストは User・Service・Device・Environment の 4 区分に分け、それぞれの関係を明確にした。理論・推論部では、モデリング部のコンテキスト記述を用いて、Java のオープンフレームワークの Jena が持つルールベース推論エンジンによって新たな知識の獲得ができる。これによってタスクの推論や通知と応答の推論が可能になった。獲得部はオープンソースライブラリの Echowand を用いて、センサやデバイスを ECHONET Lite で通信可能にし、センサ情報の取得や音声の出力ができるようにした。配信部について、ユーザは通知及び応答デバイスを予め準備した状態で、API を通じて双方向通知応答基盤を利用する。API の入力は { 通知のタイプ, 通知の開始条件, 通知の発生デバイス } となる。一方 API の返り値は { 暗黙応答, 明示応答, 手続き応答 } のそれぞれの成否 (true か false) となる。さて、双方向通知応答基盤の評価は取り扱える通知を 2 つの視点で既存の家電機器から調査し、幅広い家電製品に対して通知及び通知に対する応答が可能であることを示す。1 つ目は、最新の家電機器製品のラインナップから通知機能を割り出し、通知と応答が対応しているかを判断する。今回調査を行った 23 例の通知について、全て応答を当てはめることが出来た。2 つ目は、製品安全情報の電子機器の事故事例において、事故に至るまでの現象を分析する。そして、事故を未然に防止または事故後の事後対応を即座に行うために、事故の発生する可能性のある製品における通知と応答の組み合わせを検討する。今回調査を行った 74 の事故事例に対して、69 例について通知及び通知に対する応答を当てはめることが出来た。また、24 例については最

終現象が起こる前までに通知ができることがわかった。1つ目の評価の考察について、23例中20例の通知に対する応答は複雑な復帰応答やサービス依存応答ではなく、単純な暗黙応答もしくは現場応答となった。残りの例についても復帰応答に該当し、ユーザが指定しなければならないユーザタスク通知/サービス依存応答の例はない。このように、通常動作する家電製品の通知と応答の組み合わせを限定された種類に単純化出来たと言える。また、2つ目の評価の考察について、事故事例の74例中69例は通知と応答が可能であり、45例は緊急通知/サービス依存応答であった一方、24例についてはそれ以外の通知と応答に分類できた。このように、事故を防ぐという面において通知の多様性があるモデルと言える。以上より、双方向通知応答基盤の根幹をなす通知と応答の組み合わせは、家電製品における多様な通知と応答を単純な形で示したものである。

目次

第1章	はじめに	1
1.1	研究の背景	1
1.2	研究の目的	2
1.3	本論文の構成	2
第2章	住環境内の通知と応答について	4
2.1	通知システムの概要	4
2.2	通知に対する応答	4
2.2.1	人間とコンピュータの情報処理プロセスと相互関係	5
2.2.2	人間とコンピュータとの協調	8
2.2.3	通知インタフェースにおける人間・コンピュータ間の情報処理	9
2.3	住環境内で発生する通知とその応答の組み合わせ	11
第3章	双方向通知応答基盤の提案	16
3.1	双方向通知応答基盤の動作の流れ	16
3.2	物理空間を情報処理するためのコンテキストアウェアな基盤	17
3.3	コンテキストの獲得	19
3.4	コンテキストのオントロジモデル	20
3.4.1	User 区分	20
3.4.2	Platform 区分	20
3.4.3	Device 区分	20
3.4.4	Environment 区分	22
3.5	コンテキストの推論	22
3.5.1	タスクの推論	22
3.5.2	通知と応答の推論	23
3.6	コンテキストの配信	23
第4章	双方向通知応答基盤の実装	24
4.1	実装の概要	24
4.2	通知と応答の各種パターンの実装	24
4.2.1	現状認識通知及び動作・状態変化通知	24
4.2.2	動作終了通知、呼び出し通知、許容値外通知 a	25

4.2.3	許容値外通知 b、異常通知	25
4.2.4	ユーザタスク通知	25
4.2.5	緊急通知	25
4.3	双方向通知応答基盤 API の提供	26
4.4	プログラム上の実装	26
第 5 章	双方向通知応答基盤の評価	28
5.1	評価方法	28
5.2	最新の家電機器製品について	28
5.3	事故事例のある家電機器製品について	28
第 6 章	双方向通知応答基盤の考察	30
第 7 章	今後の課題	31
7.1	暗黙応答への幅広い対応	31
7.2	人間の行動・状態把握への対応	31
第 8 章	まとめ	32
	謝辞	32
	参考文献	34
付 録 A	製品調査	38
A.1	Echonet Lite 機器および oneM2M 機器に基づく家電機器の既存製品調査 . .	38
A.2	最新の家電機器製品の調査	41
A.3	事故事例のある家電機器製品の調査	43
付 録 B	実装例	55

目 次

1.1	プッシュ型情報による見逃し・聞き逃しの問題点	1
2.1	コンピュータと人間との間の情報のやり取り	5
2.2	Rasmussen の SRK モデル	6
2.3	2つの HIP モデル	6
2.4	人間とコンピュータが持つ機能の閉ループ及び溝	7
2.5	NA モデルでの通知に対する人間の情報処理プロセス	11
2.6	通知と応答のマトリックス	14
3.1	双方向通知応答基盤の動作の流れ	16
3.2	双方向通知応答基盤の全体像	18
3.3	コンテキストアウェアな4要素で区切った双方向通知応答基盤の全体像	19
3.4	双方向通知応答基盤のコンテキストモデル	21
4.1	双方向通知応答基盤の全体のクラス図	27
B.1	炊飯器の取扱説明書	55
B.2	通知と応答のマトリックスと、炊飯器の対応関係	56
B.3	炊飯器の炊き上がり時の動作完了通知のユースケース図	56
B.4	実機の構成	57
B.5	炊飯器の炊き上がり時の動作完了通知のシーケンス図	58
B.6	UPPAAL で用いる状態遷移モデルの凡例	59
B.7	「炊飯器の炊き上がり時の動作完了通知」の状態遷移	60
B.8	デバイスの位置情報の伝達	61
B.9	ユーザの位置情報の伝達	62
B.10	通知要求の伝達	63
B.11	通知後の応答開始要求の伝達	64
B.12	Platform から User への通知を受け取ったことの伝達	65
B.13	暗黙応答をセンサが感知した時の伝達	66
B.14	暗黙応答完了の伝達	67
B.15	現場応答をセンサが感知した時の伝達	68
B.16	現場応答完了の伝達	69
B.17	タイムアウトが発生した時の遷移	70

表 目 次

2.2	11 段階の自動化レベル	8
2.4	IRC による通知パターン	10
2.6	通知の分類	13
A.1	Echonet Lite 機器および oneM2M 機器に基づく家電機器の既存製品調査 . .	38
A.2	最新の家電機器製品における通知と応答の調査	41
A.3	事故事例から調査した製品における通知と応答	43

リスト目次

3.1	タスクの推論タスクの推論 (プレフィックスは省略)	22
3.2	通知と応答の推論 (プレフィックスは省略)	23

第1章 はじめに

1.1 研究の背景

IoTという言葉とともに、物理空間の状態をセンシングできるデバイスが急速に普及している。ユーザはセンシング情報をサービスプロバイダに送信し、有益な情報を受信できるサービスが簡単に利用できるようになった。こうした情報の送受信はスマートフォンが主であったが、ホームネットワークの普及に伴い家電機器を通して情報を送受信するサービスも多くなってきている。

情報の発信方法に注目すると、メッセージやアラートにより情報を提示する「通知」というインタフェースについて研究が行われている。ただ、通知の研究はデスクトップPCやスマートフォンを対象としたものがほとんどで、我々の身に寄り添う家電機器からの通知(例えば洗濯機の終了音、目覚まし時計のアラートなど)については製品デザイナーの領域であり、通知システムとして管理する研究は少ない。

家電機器による通知についての研究として、家電機器からユーザに向けて一方的に情報を発信するプッシュ型情報提示システム [1, 2] がある。これら研究ではプッシュ型情報をより確実にユーザに伝達するための手法を提案した。しかし、図 1.1 に示すように、情報を見逃す・聞き逃すことがあっても、それを管理システムやアプリケーションが認知できないという問題点がある。



図 1.1: プッシュ型情報による見逃し・聞き逃しの問題点

1.2 研究の目的

本研究では住環境とユーザの自然な双方向性の情報のやり取りができることを目的とし、通知と応答の組み合わせについて検討する。そして、家電製品の通知とセンサによるユーザ応答検知ができる双方向通知応答基盤 (INR: Interactive Notification-Response platform) を提案・実装する。

通知と応答の組み合わせは、既存の家電製品からどのような通知があり、どのような応答が適切かを検討する。

双方向通知応答基盤については、ホームネットワーク上の家電機器やセンサを用いて、伝えるべき情報を通知する。通知が起こった後、ユーザがどう応答したかを判断するためにはセンシング情報を処理し行動の同定をする必要がある。そこで、Echonet Lite に対応したデバイスから情報を収集し、実データをユーザ・環境・プラットフォーム・デバイスのオントロジモデルに適用して推論を行い、住環境とユーザのコンテキストをコンピュータが解釈して応答できたかを判断する。

1.3 本論文の構成

本論文の構成は以下のようになっている。

- 第1章 はじめに
 - － 研究の背景と目的、本論文の構成を示す。
- 第2章 住環境内の通知と応答について
 - － 通知および応答の概要、通知と応答の組み合わせについて示す。
- 第3章 双方向通知応答基盤の提案
 - － 双方向通知応答基盤の概要や、コンテキストウェアなサイクル、オントロジモデル、推論といった本論文で提案する手法について示す。
- 第4章 双方向通知応答基盤の実装
 - － 双方向通知応答基盤の構成、通知応答の各種パターン、API の提供といった実装に関することについて示す。
- 第5章 双方向通知応答基盤の評価
 - － 最新の家電機器製品、事故事例のある家電機器製品について調査し、双方向通知応答基盤で取り扱える通知と応答の対応関係を評価する。

- 第6章 双方向通知応答基盤の考察
 - － 通知と応答の対応関係の考察を行う。
- 第7章 今後の課題
 - － 本研究で対応できなかった暗黙応答への幅広い対応、人間の行動・状態把握への対応について述べる。
- 第8章 まとめ
 - － 本研究のまとめを記す。

第2章 住環境内の通知と応答について

2.1 通知システムの概要

通知システムとは、最新の情報を配信するためのメカニズムを用いたコンピュータシステムである。例として、デスクトップ PC・ノート PC・スマートフォンにおいて、画面の上下部から出現する 1-2 行で書かれた文字情報のポップアップや、配信情報到達時のアラーム音が挙げられる。McCrickard らの研究 [3] では、通知を次のように定義している。

複数情報への注目状態、マルチタスクを抱えている時に通常使用されるインタフェースであり、最新かつ重要な情報を、さまざまなプラットフォームやモードを通じ、効率的かつ効果的な方法で提供する。

McCrickard は設置型ディスプレイにおける通知システムの評価のため IRC(Interruption - Reaction - Comprehension) 評価モデルを提案した。評価項目は、現在進行中のタスクを中断 (Interruption) できるか、通知に対して迅速かつ正確に反応 (Reaction) できるか、通知の情報を理解 (Comprehension) し忘れずにいられるかという 3 つである。また Mehrotra らの研究 [4] では、デスクトップ及びスマートフォン上での通知が不適切なタイミングで届くときの悪影響について、これまで公開された研究を比較検討をした。結論として、通知の割り込み可能なタイミングの予測によって、ユーザが期待する心地よい通知システム構築につながるとしている。Mehrotra らは多数のデバイス (ノート PC・スマートフォン・ウェアラブル端末・スマート TV・家電製品など) による通知のモデリングは未発達な領域としている。

2.2 通知に対する応答

2.2 節では、まず人間およびコンピュータとの相互でどのように情報処理しているかを確認する。次に人間とコンピュータとの調和協調の必要性を確認する。最後に通知システムでの通知に対する応答の必要性を示す。

2.2.1 人間とコンピュータの情報処理プロセスと相互関係

コンピュータと人間との相互の情報処理はインタフェースを介してループしている。コンピュータのインタフェース装置からの出力は、光・音などによって人間が知覚する入力となる。同様に人間の応答出力は、振動・圧力・音などによってコンピュータのインタフェース装置への入力となる。つまり、Schomaker らの用いた図 2.1[5] のように、人間とコンピュータは物理媒体を通しインタフェースを出入り口として情報の受け渡しをしていると考えることができる。それでは、図 2.1 の Cognition に当たる部分はどのようなプロセスで処理するのだろうか？

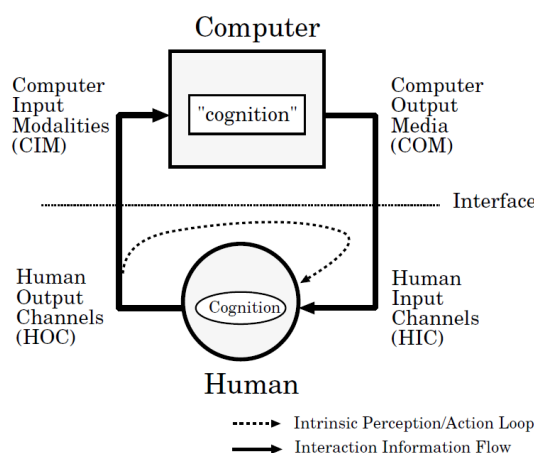


図 2.1: コンピュータと人間との間の情報のやり取り [5]

まず人間の情報処理プロセスを確認する。人間は何かしら知覚を得ると、脳内で知覚の処理を行い、応答として何かしら行動を起こす(もしくは何もしない)。Rasmussen は図 2.2 に示すように、人間の情報処理プロセスは受け取った知覚によって三段階のプロセスに変動するとし、そのプロセスを表す SRK モデル [6] を提案した。下段の SKILL-BASED BEHAVIOUR は人間が反射的な行動を起こす時のプロセスであり、人間が無意識下の身体運動を行うことを意味する。中段の RULE-BASED BEHAVIOUR は普段の習慣や規則から判断が出来る時のプロセスであり、人間が知覚情報を認識し知識や経験と照らし合わせて最適な結果となる行動をすることを表す。上段の KNOWLEDGE-BASED BEHAVIOUR は慣れない環境下で判断を迫られた時のプロセスであり、人間が知識や経験を基に到達すべき目標を決定し計画に基づいて行動をすることを意味する。

人間が何かしら行動を起こせば、物理環境またはシステム環境は変化し、人間が再び知覚可能な新しい情報パターンを生成する。このような出力が新しく入力として再帰することをフィードバックループという。Wickens らの著書 [7] では人間の情報処理は基本ステージとして知覚処理・知覚意味づけ・ワーキングメモリ・記憶保持・応答選択・応答実行を持つとした。また、Wickens らは図 2.3 に示すように、追加ステージとしてフィードバック・リソース注力の 2 つを取り入れた HIP(Human Information Processing) モデルを

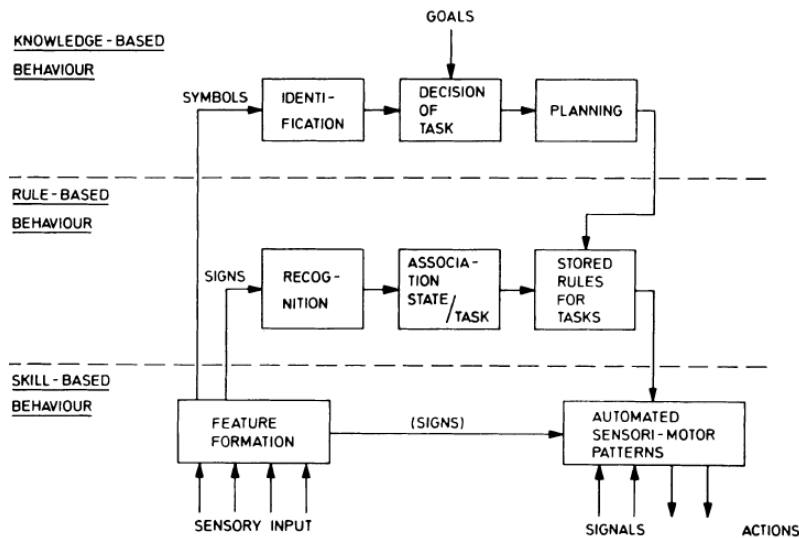


図 2.2: Rasmussen の SRK モデル [6]

提唱している。図 2.3 の左側は HIP モデルの基本形である。一方、右側は意思決定のプロセスが必要な時のモデルであり、SRK モデルの中段 RULE-BASED BEHAVIOUR 及び上段 KNOWLEDGE-BASED BEHAVIOUR と同等のプロセスを辿る。

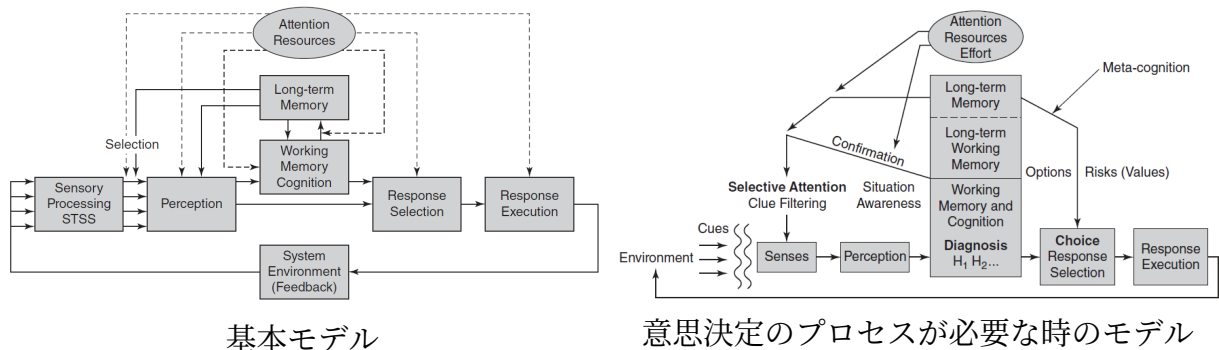


図 2.3: 2つの HIP モデル [7]

人間はフィードバック入力による知覚を受けて、現在のコンピュータが持つ情報を確認できるだけでなく、誤った行動を訂正したり意思決定を覆したりできる。Wickens は意思決定のプロセスが必要な時の HIP モデルにおけるフィードバックを 3 つに分類した。応答が行われているのと同時に並行して知覚の入力をする同時フィードバック、応答が行われた直後に知覚の入力をする時間接続フィードバック、応答に対して数秒から何か月もの間を置いて知覚の入力をする遅延フィードバックである。Wickens は 3 つのフィードバックそれぞれの特性を掴みインタフェースの設計をすることで、人間は効率の良い行動の学習ができるとしている。

次にコンピュータにおける情報処理プロセスを確認する。古来人間はプログラム言語をキーボードで打ち込み、コンピュータ上でコンパイルを実行し、望みの計算結果を画面上で確認してきた。やがてコンピュータの計算能力・精密動作が発達すると、人間は自分たちの代わりにコンピュータに意思決定および動作実行をさせるようになった。このような、人間にとって困難な作業または操作がある時、主にコンピュータが代理またはサポートとして処理を担い、人間にかかる作業負荷を軽減することを自動化という。自動化について、Parasuraman の研究 [8] ではコンピュータの情報処理プロセスとして (1) 情報の獲得、(2) 情報の分析、(3) 行動の選択と決定、(4) 行動実行、以上の 4 つの機能があると提案した。この 4 つの機能は HIP モデルが示す人間行動プロセスと似た流れを辿ることがわかる。

以上のように、人間とコンピュータの行動の処理プロセスと相互関係はこれまでの研究で定式化されており、それを踏まえてインタフェースのあるコンピュータシステムを作るのが重要である。Limerick らは人間とコンピュータの 6 つの機能が閉ループするモデル [9] を図 2.4 のように示した。図 2.4 中の Perception・Sensors が人間・コンピュータの行動プロセスの情報獲得に当たる機能、Intention・State が情報分析と意思決定に当たる機能、Effectors と Feedback が行動実行に当たる機能である。ところで、図 2.4 には Gulf of Evaluation と Gulf of Execution と書かれた矢印が存在する。これは、Norman が提案した人間と自動化との間に存在する溝 [10] を示すものである。Gulf of Evaluation はコンピュータがどのように動作しているか人間側で認識できなくなる状態を指し、Gulf of Execution は人間がコンピュータの意図通りの操作ができなくなる状態を指す。このような溝を埋めるために、自動化システムを設計する時は、人間の意思を的確にセンサで汲み取り、人間側へ動作状態・動作結果を適切にフィードバックするためのインタフェースが必要である。

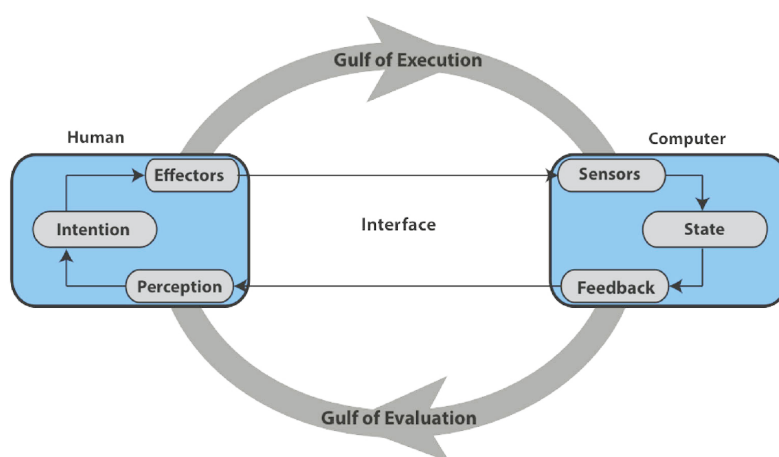


図 2.4: 人間とコンピュータが持つ機能の閉ループ及び溝 [9]

2.2.2 人間とコンピュータとの協調

自動化の技術が発達することで、かつて人間が単独で行っていた作業を現在はコンピュータと共に連携しながら作業を進められるようになってきた。Sheridan らは 10 段階の自動化のレベル [11] を表現し、人間とコンピュータのどちらがどのようにタスクを実行すべきか幅があることを示した。また、稲垣らは Sheridan の 10 段階の自動化の内、レベル 6 と 7 の間に新しいレベルを挟んだ 11 段階の自動化レベルを表 2.2 のように提示し、レベル 6.5 を人間の対応の実行までに時間の猶予がない緊急時に必要な段階であると主張した [12]。

レベル	説明
1	コンピュータの支援なしに、すべてを人間が決定・実行。
2	コンピュータはすべての選択肢を提示し、人間はそのうちの一つを選択して実行。
3	コンピュータは可能な選択肢をすべて人間に提示するとともに、その中の一つを選んで提案する。それを実行するか否かは人間が決定。
4	コンピュータは可能な選択肢の中から一つを選び、それを人間に提案。それを実行するか否かは人間が決定。
5	コンピュータは一つ案を人間に提示。人間が了承すれば、コンピュータが実行。
6	コンピュータは一つ案を人間に提示。人間が一定時間以内に実行中止を指令しない限り、コンピュータはその案を実行。
6.5	コンピュータは一つ案を人間に提示すると同時に、その案を実行。
7	コンピュータが全てを行い、何を実行したか人間に報告。
8	コンピュータが全てを決定・実行。人間に問われれば、何を実行したか人間に報告。
9	コンピュータが全てを決定・実行。何を実行したか人間に報告するのは、必要性をコンピュータが認めた時のみ。
10	コンピュータが全てを決定し、実行。

表 2.2: 11 段階の自動化レベル [12](日本語訳 [13])

人・時間・場所・環境が変われば、人間とコンピュータのタスク分担も変わっていかなくてはならない。このような、人間とコンピュータの周辺環境も含めた状況に合わせ、タスクの移譲ができるように調整していくことをアダプティブオートメーション (Adaptive Automation) という [14, 15, 16]。例えば、様々な家電機器をシステムが一元管理しているスマートホームがあり、室内温度が人間にとって危険な程高いと判断した時に「室内が高温です。窓を開けましょう。」という音声を通知をしたい場合を想定する。自動化レベル 4 の場合から考察すると、この場合コンピュータは通知後に何もしない。レベル 5 の場合は、人間にスマートフォンで自動開閉窓のオート操作ボタンをタップしてもらい、タップがあれば最適な温度になるよう自動で動く。レベル 6 の場合は、人間にスマートフォンでオート操作を拒否するボタンのタップを一定時間待ち、タップがなければ最適な温度になるよう自動で動く。ここまでは人間が意思を持って状態の改善をするかどうか選ぶことが出来たが、レベル 6.5 以上は人間が自動開閉窓の動作開始を止めることが出来ない。そのためレベル 6.5 以上を考慮すべき段階は、温度が人体に危険な水準まで上昇した場合のよ

うな限定した状況でなければならない。以上のような様々な想定をするために、設計者は自動化の対象について事前に重要度や緊急度について調査をし、どのレベルが適切か検討を重ねる必要がある。

人間とコンピュータとが相互に信頼を築くためには、コンピュータが人間のようになりきって振る舞うことで、人間と人間が会話をしているかのように思わせる必要がある。Nass らの研究 [17] では、人間はコンピュータに対して恰も人間に接するかのように意思疎通することを示した。協調の取れた自動化を実現するためには、コンピュータは人間の仕草を理解し、コンピュータ自身の動作に納得してもらう必要があるのだ。

2.2.3 通知インタフェースにおける人間・コンピュータ間の情報処理

McCrickard の IRC 評価モデルの観点で、通知というインタフェースについて人間がどのように情報処理するのか確認する。McCrickard の研究 [3] では、2.1 節で述べた IRC 評価モデルによって分類した通知パターンが、Wickens らの HIP モデル [7] の情報処理プロセスにどう適用できるか見解を示した。まず IRC による通知パターンについて表 2.4 に示す。それぞれのパラメータについて、I(中断) のパラメータは情報提示によって作業を中断する必要があるかどうか {1=中断する, 0=中断しない}、R(反応) は直ちに応答する必要があるか {1=直ちに応答, 0=直ちに応答しなくてもいい}、C(理解) は得られた情報が複雑な意思決定をする時に役立つか {1=役立つ, 0=役立たない} を表す。なお、これらのパラメータは 0 か 1 か二極しているわけではなく、0-1 の間に通知が存在する。本研究ではわかりやすさのために、0 か 1 の値を用いる。パラメータによる NA モデルの例を示す。Ambient Media のデスクトップ背景が天気によって動的に変わるという例であれば、I(中断) が 0 なのでデスクトップ背景に意識を取られず、R(反応) が 0 なので天気の情報に直ちに応答を起こさなくてよく、C(理解) が 1 なので情報から外出時に傘がいるかどうかを判断する必要がある。また Alarm のメール到着時のアラートであれば、I(中断) が 1 なのでポップアップを見るために仕事を一時中断し、R(反応) が 1 なので直ちにメールを確認するが、C(理解) が 0 なのでポップアップ自体はメールソフトにメールが届いたという示唆に留まる。

McCrickard は IRC 評価モデルによる通知パターンを HIP モデルに適用した NA モデル (Notification Action Model) を図 2.5 のように示し、通知における人間の情報処理プロセスを表した。まず、8 つの通知パターン内にある HIP モデルの機能ブロックについて説明する。SP(Sensory Processing) は感覚処理を表し、視覚、聴覚、触覚といった感覚を処理する。P(Perception) は知覚を表し、感覚信号やイベントを意味づける。WM(Working Memory) は作業記憶を表し、長期記憶に情報を送ったり、情報認識の処理をしたりする。LTM(Long Term Memory) は長期記憶を表し、過去の経験を保存する。LTWM(Long Term Working Memory) は長期作業記憶を表し、長期に渡る情報認識の処理を行う。RS(Response Selection) は応答選択を表し、最終的に何をするかを定める。RX(Response eXecution) は応答実行を表し、脳指令を含めた筋肉運動を行う。次に、通知パターンのプロセスを確認す

パターン名	Interruption 中断	Reaction 反応	Comprehension 理解	例
Noize (ノイズ)	0	0	0	作業中にインターネットラジオを聞くことで安心できる。
Ambient Media (環境に溶け込んだ メディア)	0	0	1	パソコンのデスクトップ背景で天気に関する画像を動的に表示し、外の天気を気づかせる。
Indicator (標識)	0	1	0	カーナビのディスプレイ上にある方向指示案内に従って進む。
Secondary Display (二つ目のディスプレ イ)	0	1	1	共著者の進捗度をサブモニタ上のグループウェアで確認しながら仕事をする。
Diversion (気晴らし)	1	0	0	時々スマートフォンにポップアップするジョークボットで気晴らしする。
Information Ex- hibit (情報表示)	1	0	1	工場で操業のステータスをディスプレイ画面で日常的に確認、短期的な情報の更新を見つつ、長期的な戦略を決める。
Alarm (アラーム)	1	1	0	カレンダーやメールのアラートで重要な情報が発生したことを認識する。
Critical Activity Monitor (要所の 活動を監視)	1	1	1	ネットワークを日常的に監視する。ユーザは監視者に問題を解決してもらう。監視者は問題をパターン化し、問題を事前に防いだり問題の発生しないネットワークを構築する。

表 2.4: IRC による通知パターン [3]

る。それぞれのパターンでは共通したプロセスを辿る場合があり、SP の後は P、RS の後は RX、RX の後は F に進む。いくつかのパターン例を挙げると、Ambient Media の場合は SP または P の後に LTWM に進むが、その後に F に進むかどうかは重視されない。他に、Alarm の場合は P の後に WM に進み、WM の後に RS に進んでいく。NA モデルでは、IRC のパラメータによってどの機能ブロックを通るかおおよそ把握できる。I=1 の時は WM を通り、R=1 の時は RS・RX を通り、C=1 の時は LTM・LTWM を通る。

通知に対して人間がどう応答するかを把握するのは重要である。2017 年の Turner らの研究 [18] では、多くの通知割り込みの実践研究において、実験後にアンケート調査で通知が来た時どう感じたかを尋ねたり、通知到着時にその場でユーザが応答可能かラベリングを行ったりしているという見解を述べた。前者はコンピュータが応答をセンシングしておらず、後者はスマートフォンによりセンシングしているがより多くの情報獲得が可能である。しかし、2.2.1 及び 2.2.2 小節で挙げたことを考慮すると、通知インタフェースにおける人間・コンピュータの情報処理プロセスを理解し、通知デバイス(コンピュータから人間へのインタフェース)と応答センシング(人間からコンピュータへのインタフェース)が機能すれば、人間は通知という声を通してコンピュータと協調してタスクを行えるようになる。そのため、センシング技術の発達に合わせて通知に対する応答センシングをより探求する必要がある。

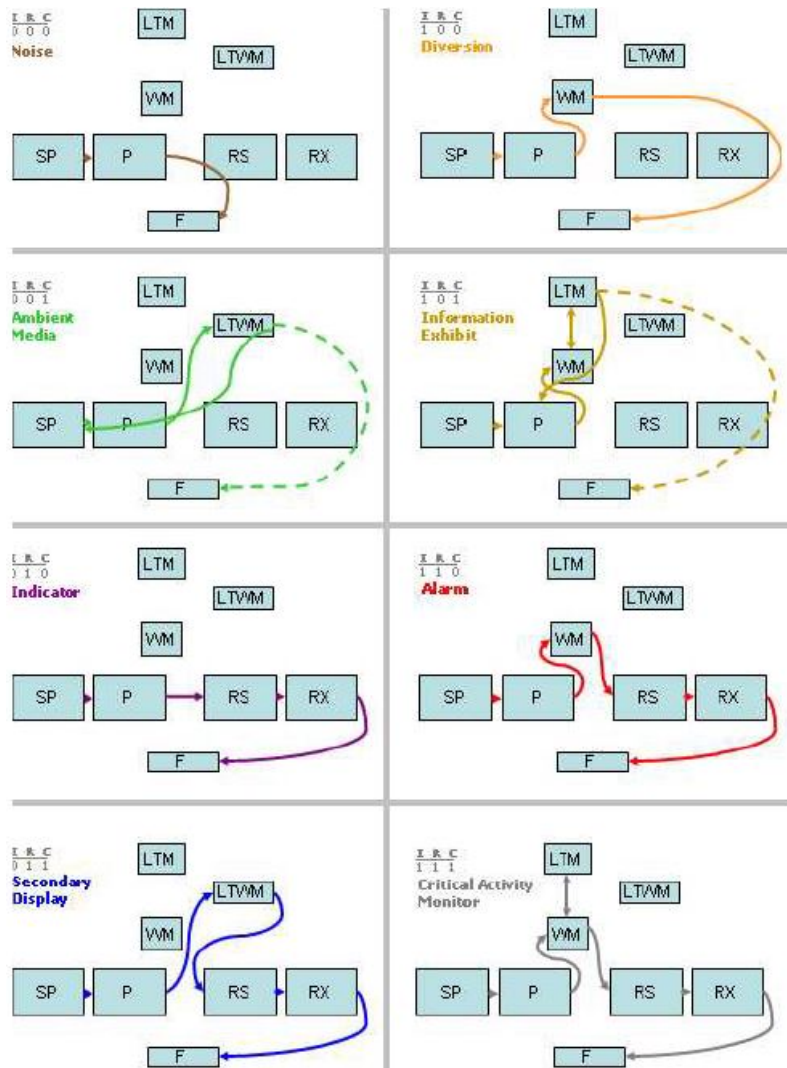


図 2.5: NA モデルでの通知に対する人間の情報処理プロセス [3] (図は [19] より)

2.3 住環境内で発生する通知とその応答の組み合わせ

HCI(Human-Computer Interaction) の分野でデスクトップ PC やスマートフォンにおける通知の研究がなされてきた一方、近年普及が進む IoT デバイスでの通知については 2.1 節の Mehrotra の言う通り研究余地のある分野と言える [4]。

福田, 金井の研究 [1, 2] では、インターネットから入手した情報を一方的に家庭内の家電機器からユーザに向けて発信するプッシュ型情報を提案し、その情報をより確実にユーザに伝達する方法を提案した。しかし、通知デバイスからユーザに対して単方向に通知するため、通知を見逃す・聞き逃すことがあっても、それをアプリケーションが認知出来ないという問題点がある。

Teodoro らの研究 [20] では、多数のデバイス上での通知を可能にする XDN(Cross-device Notification) Framework という開発者サポートツールの実装を行った。開発者はデバイスの対象として { スマートフォン, スマートウォッチ, スマートブレスレット, スマートライト, タブレット, PC, スマート冷蔵庫, スマートハイファイオーディオ, スマートTV, カーハイファイ, スマート歯ブラシ } を選び、API を用いることでデバイスの持つ音・光・振動機能による通知ができる。このようないくつかのスマート家電を組み合わせた通知システムは他にも研究例がある。

しかし、家電機器のスマートデバイス化が進む今日、本研究筆者は Teodoro らの研究が対象とするデバイス種類より多くの通知を扱うべきであると考え。ホームネットワーク上にある様々な機器と通信するためのプロトコルである ECHONET LITE[21] は、家電製品の対応機器 100 種以上を定義している。この中で音声・アラーム音の通知が可能なデバイスは { 家庭用エアコン, 電気温水器, 電気錠, 瞬間式給湯器, オーブンレンジ, 冷蔵庫, 炊飯器, 洗濯機, 乾燥機 } などがある。これらデバイスを包含した通知を単一のシステムを通して扱えるようになれば、日常生活で発生する報知音やディスプレイ通知といった「デバイスの発する呼びかけ」に耳を傾けやすくなる。そして、応答として人間の行動をセンシングすることでコンピュータと人間との間の情報のやりとりを循環できるようにする。

ECHONET Lite 機器 [22]・oneM2M 家電機器 [23] のクラスより、音声・視覚的シグナルを発する 17 機器を調査し、A.1 節の表 A.1 にまとめた。調査ではインターネット上のメーカ説明書より通知機能を合計 46 個特定した。こうした通知について、{ メーカー名, 製品名, 通知場所, 通知トリガ, 通知方法, 通知デバイス, 通知種類, 応答種類, 応答デバイス } といった項目で分析した。通知について { 1. 現状認識, 2. 動作・状態変化, 3. 動作終了, 4. 呼び出し, 5a. 許容値外 a, 5b. 許容値外 b, 6. 異常, 7. タスク, 8. 緊急 } の 9 種類に分類した。また、応答について { A. 暗黙応答, B. 明示応答, C. 手続き応答 } の 3 つの大区分と { B1. 現場応答, C1. 復帰応答, C2. サービス依存応答 } の 3 つの小区分に分類した。

まず、通知の分類を表 2.6 にまとめた。それぞれの通知種類について説明する。1. 現状認識通知は、時計の時報のような定期的に行う通知を表す。2. 動作・状態変化通知は、予め設定した動作が行われたときやコンピュータの判断により自動で動作を変更した時の通知を表す。1. と 2. は直接人間がタスクを実行して解決するわけではないが、今後の意思決定に役立つ。3. 動作終了通知は、コンピュータが行っていたタスクが完了し、その後のタスクを人間に引き継いでもらうための通知である。4. 呼び出し通知は、人間が早急に対応しなければならないタスクをコンピュータが呼びかけるための通知である。5a と 5b の許容値外通知は、コンピュータ内で許容値を超える何かしらの条件に当てはまる時に人間を呼び出すための通知である。5a と 5b の違いについて、5a はタスク解決が人間の主体的な一操作で完結する時で、5b は一動作で完結せず追加の操作や時間の経過が必要な時の通知とした。6. 異常通知は、コンピュータ・デバイス内でエラーが発生した時の通知である。7. ユーザタスク通知は、ユーザが独自に設定し、他の通知種に当てはまらない時の通知である。8. 緊急通知は、ユーザの生命・財産に害がある時の緊急性の高い通知である。

ID	通知種類	タスクとの関係	タスクにか かる時間	NA モデル (IRC)	緊急度
1	現状認識通知	直接タスクと関係ない	無	Information Exhibit(101) → Ambient Media(001)	無
2	動作・状態変化通知	タスク変更・タスク解決後の通知	無	Information Exhibit(101) → Ambient Media(001)	無
3	動作終了通知	既存タスク解決後、新規で人間が行うタスクが発生	短	Alarm(110) → Indicator(010)	無
4	呼び出し通知	新規で人間が行うタスクが発生	短	Alarm(110) → Indicator(010)	無
5a	許容値外通知 a	新規で人間が行うタスクが発生	短	Alarm(110) → Indicator(010)	無-低
5b	許容値外通知 b	新規の異常状態タスク発生	長	Critical Activity Monitor(111)	低
6	異常通知	新規の異常状態タスク発生	長	Critical Activity Monitor(111)	中
7	ユーザタスク通知	新規のユーザ設定タスク発生	短-長	全て	無-高
8	緊急通知	新規の緊急状態タスク発生	長	Critical Activity Monitor(111)	高

表 2.6: 通知の分類

次に、応答の分類について説明する。A. 暗黙応答とは、人間の無意識的な行動であり通知を認識した直後に通知デバイスを見たり、デバイスに近寄ったりするような応答である。B. 明示応答とは、過去の経験と学習によって獲得したルールによる意識的な行動で、暗黙応答の後に何かしらのデバイスの操作をする応答である。通知が行われたデバイスに移動し、そのデバイスに接触して操作をする応答を特に B1. 現場応答と呼ぶ。C. 手続き応答とは、目的達成に向けて状況を判断する必要のある行動で、暗黙応答、明示応答の後に行われる高度な作業を伴う応答である。窓を開け、時間をかけて部屋の温度が下がったことを温度センサに検知させたり、故障したデバイスを修理したりするといった、デバイスの状態やユーザの身の回りの環境を正常に戻す応答を特に C1. 復帰応答と呼ぶ。他に、照明が点滅したらリビングで食事ができたことを表すといった事前のルールが必要な通知や、持ち物に事前に電子タグをつけて玄関で忘れ物をチェックする通知、火災報知や防犯警報といった通知など、実現したい通知によって複雑な手順を踏む通知に対する応答を C2. サービス依存応答と呼ぶ。

そして、通知と応答の対応関係をユースケースと共にマトリックスで記述したものを図 2.6 に示す。行の並びは重要度順に並んだ通知である。上から下にかけて重要度が高くなる。3 列目から 5 列目の列の並びは応答の積み重なりを表す。左から右にかけて、応答の早い順となる。

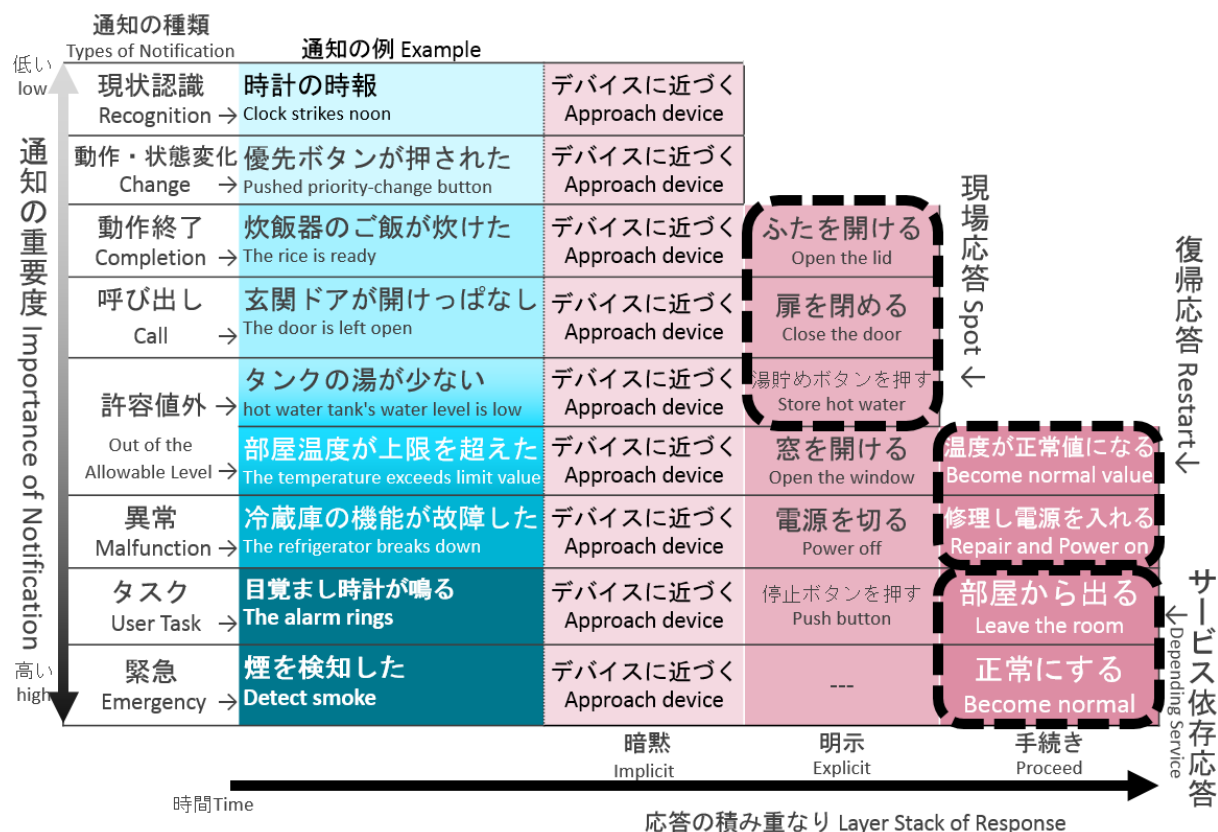


図 2.6: 通知と応答のマトリックス

図 2.6 中にはそれぞれの通知種に対応する応答種があるが、通知の重要度が高くなればなるほど応答の積み重なりも増えていく。このような対応関係は、表 2.2 の自動化におけるタスクとの関係、図 2.5 の NA モデルの情報処理プロセス、及び通知の緊急度を元にして、開発者が通知種を判断しやすいようにバランスを取って設計した。順を追って対応関係を説明する。

1. 現状認識通知は、人間が行うべきタスクを提示するわけではないので反応する必要がない (R=0) が、その情報自体は今後の意思決定に役立つ (IRC=101 → 001)。従って、NA モデル (Information Exhibit → Ambient Media) より応答については重要視しなくても良いため、暗黙応答のみをセンシングする。2. 動作・状態変化通知は、タスク変更・タスク解決後の通知を行うため、反応する必要はない (R=0)。コンピュータが行ったタスクの認識のため 1 より重要度が高いが、タスクを担うわけではない。そのため、NA モデルは 1 同様 (Information Exhibit → Ambient Media) とし、暗黙応答のみをセンシングする。なお、1 と 2 が (IRC=101 → 001) や NA モデル (Information Exhibit → Ambient Media) のように矢印が付くのは、人間の慣れによって中断 (Interruption) が起こりにくくなるためである。

3. 動作終了通知は、人間が設定した既存タスク解決後に新規で人間が行うタスクが発

生する。そのため、NA モデルは反応が必要である ($R=1$) が、タスクを行うべき場所で簡単な行動をすればタスクは終わるので、タスクへの深い理解は必要ない ($C=0$)。従って、NA モデル (Alarm $\rightarrow$ Indicator) より応答は暗黙応答を行った後、タスクを行ったかどうかの明示応答について判定を行う。この時の応答は通知が発生した場所で短時間のタスクを解決することから、現場応答という名前とする。4. 呼び出し通知は、人間の予期が難しいタイミングに起こる新規のタスクが発生する。他の考慮点は3と同じで、応答種は現場応答である。4の方がタスクの移譲が難しいため、重要度を高くした。5a. 許容値外通知 a は、新規のタスクが発生する。この場合は3や4と比べてコンピュータが許す正常値を上回ったり下回ったりするが、人間に危害がほとんど無いタスク発生のお知らせである。これも人間に一度操作をして状態を回復できるため、応答種は現場応答である。なお、1や2と同様、3・4・5も慣れが発生し中断 (Interruption) が起こりにくくなるので、($IRC=110 \rightarrow 010$) や NA モデル (Alarm $\rightarrow$ Indicator) のように矢印が付く。

5b. 許容値外通知 b は、5a に比べて緊急度が高くタスクに時間を要する時の通知である。コンピュータが許す正常値を大きく上回ったり下回ったりする場合を想定し、その回復に時間がかかるため、暗黙応答、明示応答の後にタスクが解決されたかチェックするための手続き応答を必要とする。日常生活であまり起こらないことを考慮し、中断に慣れない $IRC=111$ で、NA モデルは Critical Activity Monitor となる。6. 異常通知は、デバイスが強制終了したり故障してしまった時のための通知である。その影響は通常の動作が出来ない分、5b より重要度が高い。5b と 6 はその異常を復帰するタスクが生まれ、対応に時間がかかるので応答は3段階で、3段階目の手続き応答を復帰応答という名前とする。

先に8を説明する。8. 緊急通知は、人体や財産に危機が迫っているため、最も重要度が高い通知とした。人間は臨機応変に対応をする必要があるため、応答はいくつか手順を踏むと考え、応答の積み重なりを3段とした。このように応答はとっさの動作であり、タスクに依存して人間の応答が大きく変動すると思われ、サービス依存応答という名前とする。

それでは7を説明する。7. ユーザタスク通知は、ユーザが自由に通知応答を決めることができる時の通知である。目覚ましで起きた後に確実に家を出るまでを応答とするようなサービスを考えると、その応答パターンはこれまでのものでは対応できない。そのため、8と同様サービス依存応答となる。ただし、緊急通知という直感的に重要だとわかるものを最高の重要度にしたかったため、8のひとつ下の通知を7のユーザタスク通知とした。

第3章 双方向通知応答基盤の提案

3.1 双方向通知応答基盤の動作の流れ

本研究では「双方向通知応答基盤: Interactive Notification-Response(INR) platform」を提案・実現し、住環境とユーザの自然な双方向性の情報のやり取りをするという目標を達成する。双方向通知応答基盤の全体の動作の流れを図3.1に示し、順を追って説明する。

1. ユーザはアプリケーションに通知を設定する。
2. アプリケーションは双方向通知応答基盤に通知要求を出す。
3. 双方向通知応答基盤は受け取った通知要求を元に、ホームネットワーク上の家電機器に通知命令を送る。
4. 家電機器は通知命令を受け取ると、ブザー音・音声案内・視覚的シグナルといった通知を出す。
5. ユーザは通知を認識し、通知に従った行動を起こす。
6. センサはユーザの行動を応答として様々なセンサを用いてセンシングする。
7. 双方向通知応答基盤はセンサから応答情報を受けとり、内部でアプリケーションの通知要求に対して正しい応答であるかを判断する。
8. 双方向通知応答基盤は通知の成否をアプリケーションに返答する。

なお、複数回通知をする場合、及びユーザが複数の応答をする場合があるため、一度の通知要求において3,4の情報の流れ、及び5,6,7の情報の流れが複数回発生する場合がある。

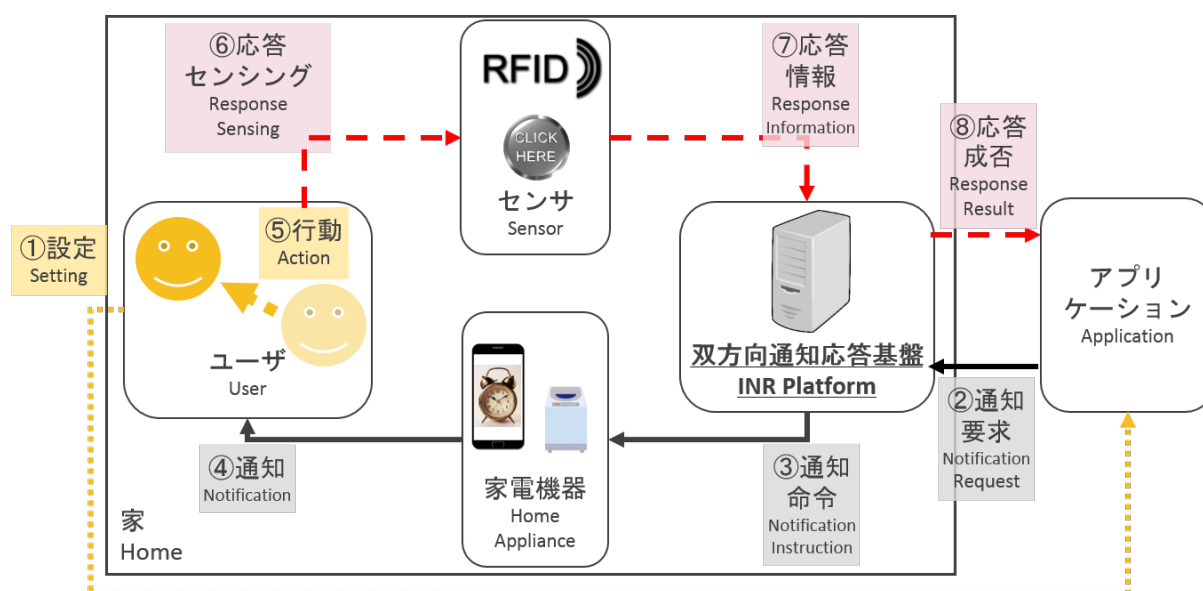


図 3.1: 双方向通知応答基盤の動作の流れ

3.2 物理空間を情報処理するためのコンテキストウェアな基盤

いつ、どのデバイスが、どんな方法で、どのユーザに通知し応答を取得すべきかは、ユーザ・デバイス・ユーザ周りの環境といった幅広いコンテキストから総合的に判断すべきである。3.2節では双方向通知応答基盤がどのようにコンテキストを処理し、目標を達成するのかを提示する。

Perera らは、コンテキストウェアな情報処理をするために、獲得、モデリング、理論・推論、配信という4要素によるコンテキストライフサイクルを定義した[24]。まず獲得部では、物理または仮想センサによる様々な情報を得る。次にモデリング部では、獲得部で得られる情報を意味のある表現形式にする。また理論・推論部では、低レベルの生センサデータを高レベルのコンテキスト情報に変換する機能を持つ。そして配信部では、ユーザが必要とする低レベル・高レベルどちらの情報も提供する機能を持つ。以上のように、コンテキスト情報を要素別で管理する方法は、効率的かつ効果的なコンテキストウェアのシステムの構築に役立つ。

Chen らの研究[25]では、コンテキスト共有モデルの維持及び管理ができる Context Broker を導入した CoBrA(Context Broker Architecture)を提案した。Context Broker は、(i) 多種の情報源からコンテキストを獲得し、推論を通してコンテキスト全体の一貫性を維持する、(ii) オントロジー、エージェント及びプロトコル間の通信によりアーキテクチャ内に分散する機能がコンテキスト知識を共有できるようにする、(iii) 重要な個人情報アーキテクチャ内の機能と共有しながら、ユーザ定義のポリシーを設定・運用することで、ユーザのプライバシーを保護する、という以上3つの責務を果たす。Context Broker を適用した研究として、Hervas らの COIVA[26]がある。アーキテクチャのコア部分に Context Broker を配置し、OWL で構築したユーザ・環境・デバイス・サービスのオントロジモデルや、問い合わせ言語 SPARQL によるコンテキスト推測エンジンといった機能との仲介の役割を果たした。これにより、様々なサービスに対して統合的なコンテキストによる情報提示ができるようになった。

本研究では Perera のコンテキストライフサイクルを構成する4大要素と、Chen らの Context Broker を双方向通知応答基盤に取り入れ、ユーザを取り巻く環境を情報処理する。図3.2は双方向通知応答基盤の全体像である。中央に Context Broker(図3.2中では単に Broker と書かれた部分)を配置しており、Distribution API・Context Manager・Device Manager といった全体の機能との仲介をする。

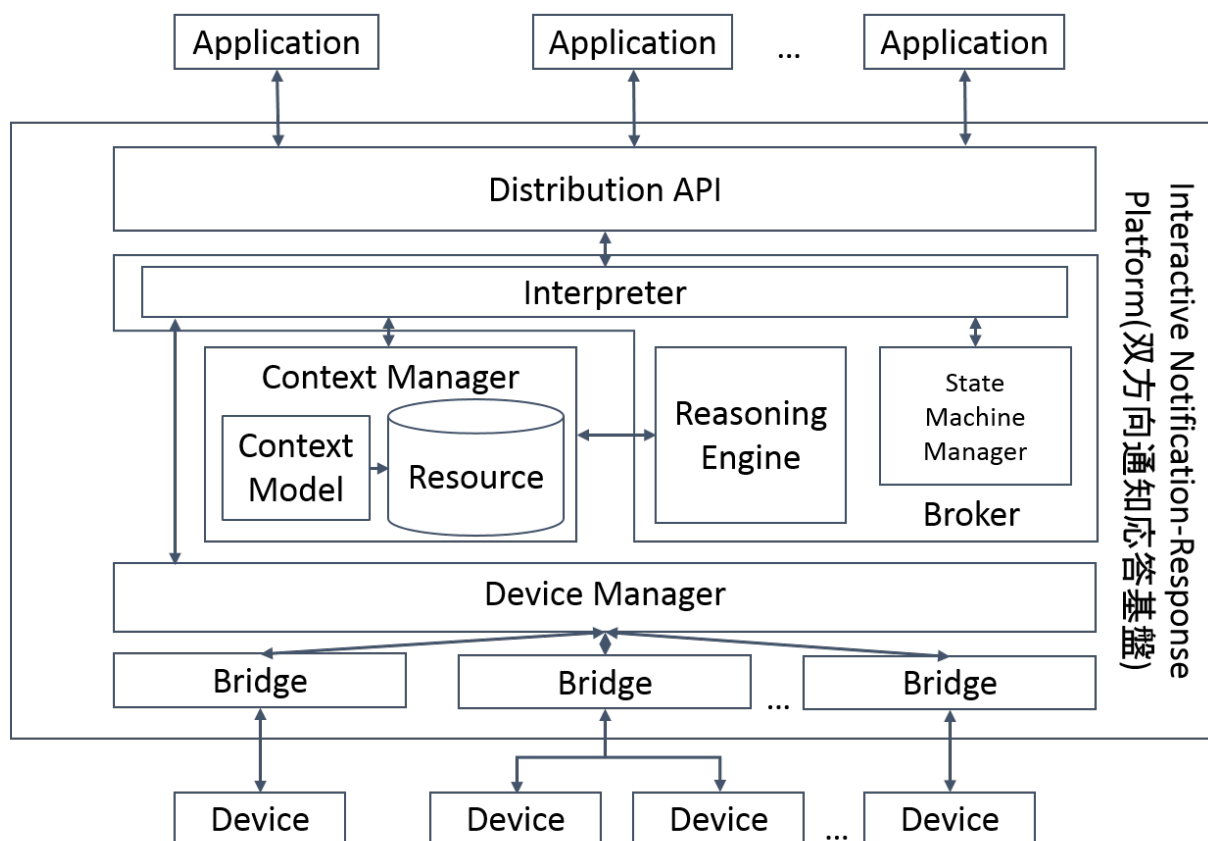


図 3.2: 双方向通知応答基盤の全体像

コンテキストアウェアな4要素を含んだ双方向通知応答基盤の全体像を図3.3に示す。図の上部がコンテキストの配信、中央部分がコンテキストのモデリング、中央右部分がコンテキストの理論・推論、中央左及び下部がコンテキストの獲得とそれぞれ対応している。Brokerの内、Interpreterは4要素それぞれの情報の橋渡しの役割を果たす。配信部のDistribution APIはApplicationに対して双方向通知応答基盤に接続するためのインタフェースを提供する。理論・推論部について、Reasoning Engineでは問い合わせ文や知識を生成するためのルールを管理し、State Machine Managerは実時間に合わせたデバイスや人間の状態遷移を管理し通知の成否を判断する。モデリング部(Context Manager)では、Context Modelで表現したデータ形式のResourceに対して推論を行うことでデータを引き出したり、新しいデータを作ったりする。獲得部について、Device ManagerはBridgeに対してDeviceで通知するための命令を出したり、Bridgeを介してDeviceから応答を獲得したりするための管理機能を持つ。

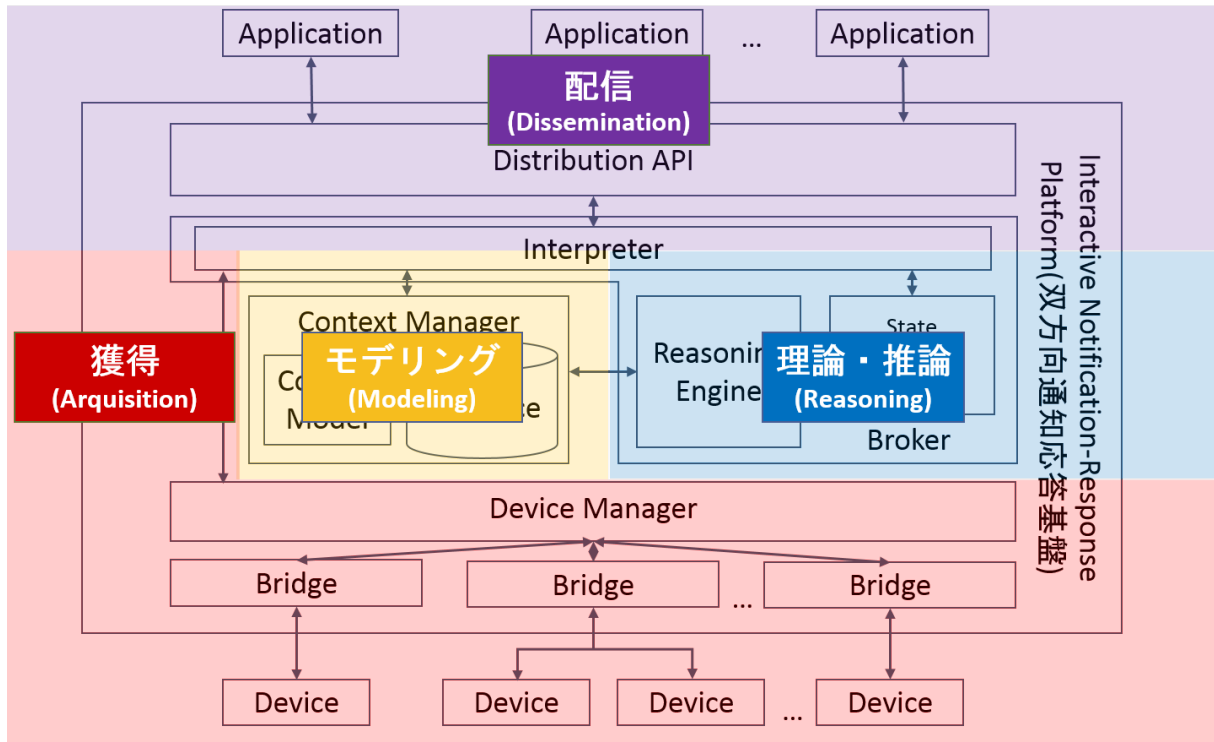


図 3.3: コンテキストウェアな 4 要素で区切った双方向通知応答基盤の全体像

本章では 3.3 節で獲得部、3.4 節でモデリング部、3.5 節で理論・推論部、3.6 節で配信部について詳細を説明する。

3.3 コンテキストの獲得

物理センサによって獲得すべき情報は、ユーザの GPS 位置情報・デバイスの稼動状態情報・応答用のセンサ情報など様々であるが、本研究ではユーザが屋内に居ることを想定し、スマートハウス内の家電機器制御を果たす共通のプロトコルである Echonet Lite によりコンテキストの獲得を行う。Pham らの研究 [27] では、Echonet Lite の仕様書に合わせたオントロジモデルを提案し、Echonet Lite Adaptation Layer によって実デバイスと RDF フォーマットのリソース形式との通信を可能にした。本研究上では家電機器が Echonet Lite に対応していることを想定しており、Github 上にある Pham らの Echonet Lite Adaptation Layer[28] を活用した実装を行う。しかし、コンテキスト獲得時に用いるオントロジは OWL フォーマットで統一してあれば相互運用が可能であるため、位置情報や家屋情報などを OWL フォーマットで別途定義したり、既存のオントロジを活用することもできる。例えば、IoT プラットフォームの oneM2M で定義されたリソースを OWL フォーマットとして記述した The oneM2M Base Ontology[29] があるが、これに対応したデバイスのリソース表現も可能である。

3.4 コンテキストのオントロジモデル

双方向通知応答基盤におけるユーザとデバイス周辺の環境についてのコンテキストを OWL で書き表す。図 3.4 はビジュアルエディタ WebVOWL[30] 上でコンテキストを表す OWL を表示したものである。User・Service・Device・Environment で分けた 4 つの区分に従って、それぞれのクラスを説明する。

3.4.1 User 区分

User は双方向通知応答基盤を使用するユーザを表し、目的語として App・Goal・Task・Environment を持つ。User は双方向通知応答基盤を用いるアプリケーションを使用し (use-App)、双方向通知応答基盤を通して達成したい目標を持ち (aim)、達成に必要なタスクに関わり (InvolvedIn)、いつも室内外いずれかの環境に属している (locatedIn) ことをそれぞれ示す。Goal は人間が規定する目標を表し、目的語として Linker・Genre・Trigger・AimObject・Device・Task を持つ。Goal は双方向通知応答基盤が管理する通知と応答の関連付けと結びつき (hasLinker)、通知のタイプ・通知の開始条件・通知の発生デバイス・ユーザが解決すべきタスク (hasGenre・hasTrigger・hasAimObject・hasTask) を持つ。Genre はタスク生成時に必要な通知のタイプ情報を持つ。Trigger は通知の開始タイミングを知るための情報を持つ。AimObject は通知が発生するデバイス情報を持つ。Task は目標達成するために必要な人間の動作一つ一つを表現し、目的語として Device を持つ。Task は応答するためにデバイスを使用する (useDevice)。

3.4.2 Platform 区分

App は双方向通知応答基盤を使用する開発者でない一般ユーザが使うアプリケーションを表し、目的語に双方向通知応答基盤 Platform・Goal を持つ。App は双方向通知応答基盤に依存し (useINR)、目標を実現する (realize)。INRPlatform は双方向通知応答基盤全体を表し、目的語として Linker を持つ。INRPlatform は通知と応答を関連付ける機能を管理する (manage)。Linker は通知に対する応答がどんなものであるかを結び付きを表し、目的語として Notification・Response を持つ。Linker は通知と応答がどんなものがあるかという情報を持つ (hasNotification・hasResponse)。Notification と Response は通知と応答を表し、目的語として Device を持つ。Notification と Response はデバイス进行操作して (control) 通知と応答を行う。

3.4.3 Device 区分

Device は EchonetLite のオブジェクトを参考にしたデバイス群を表し、目的語に Environment を持つ。Device は住環境に属している (locatedIn)。EchonetLite の OWL による

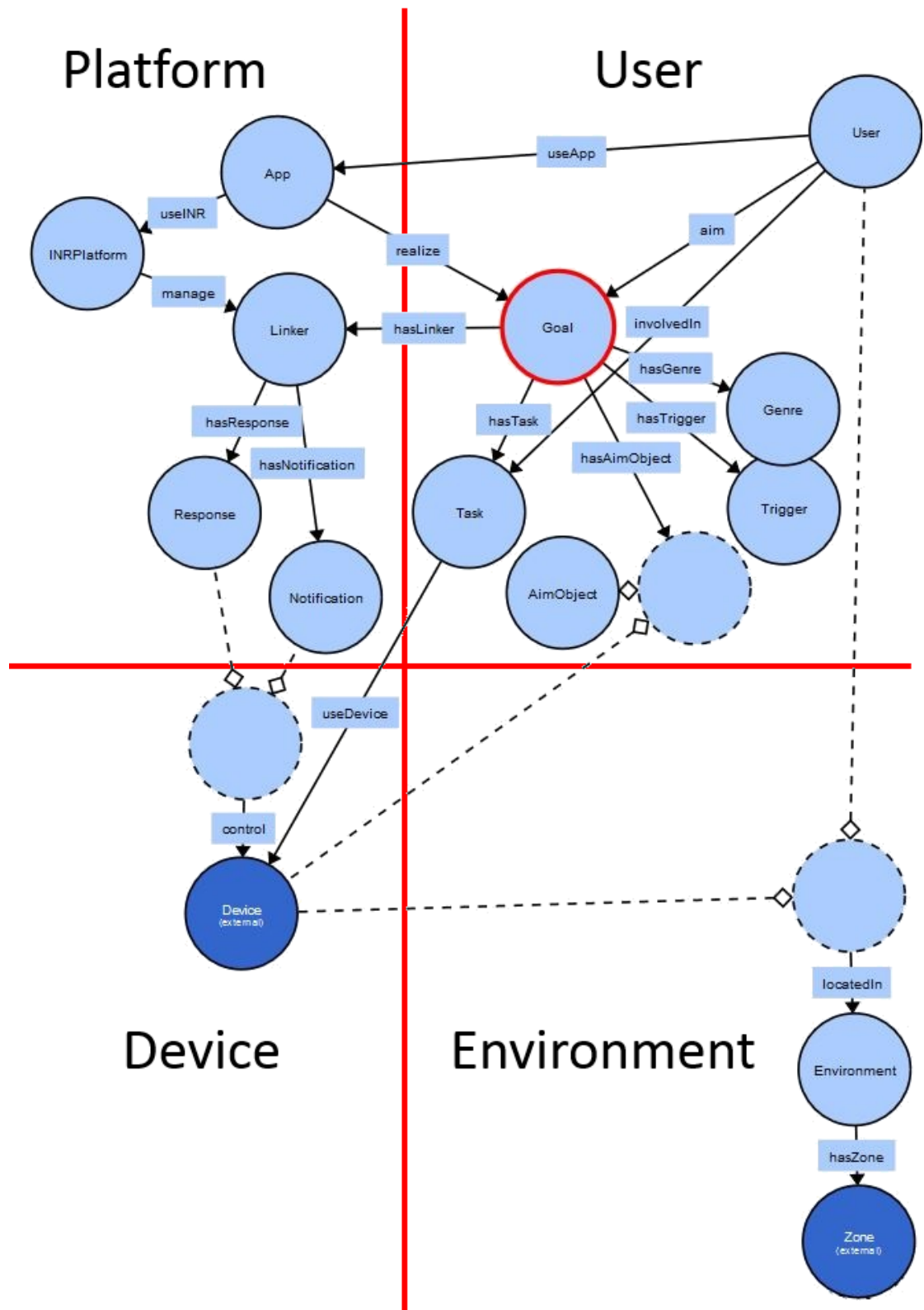


図 3.4: 双方向通知応答基盤のコンテキストモデル

記述は Pham らの Echonet Lite Adaptation Layer[28] をインポートしている。

3.4.4 Environment 区分

Environment は住環境を表し、目的語に Zone を持つ。Environment は属する場所を持つ (hasZone)。このオントロジ内の Zone は、スマートホーム環境を OWL で記述した The SEAS Building Ontology[31] の Zone 及び下位クラスをインポートしている。SEAS Building Ontology では建物、部屋、物理環境といった情報を扱っている。

3.5 コンテキストの推論

本研究では OWL で記述した概念を個体記述するために RDF を用いているため推論が可能である。これによって情報同士の関係を理解している人間がルールを与え、新たなデータを得ることができる。本研究では OWL フォーマットを解釈し、リソース情報を蓄積・変更・削除などができる Java のオープンフレームワーク Jena[32] を用いる。Jena の持つ推論エンジンにおいては、(前提条件 1),(前提条件 2), $\dots \rightarrow$ (結論 1),(結論 2), $\dots$ といった形で新たなデータの獲得ができる。なお、本来は「`inr:Goal`」のように、どの OWL による概念かを示すプレフィックス部が必要であるが、本論文では見やすさのために省略してある。State Machine Manager にあたる状態遷移部については、4 章で説明する。

3.5.1 タスクの推論

ある通知に対する行うべき行動の目標を与えられたら、そこから人間が行うタスクを推論する。リスト 3.1 のルールでは、目標の 3 つの要素 (ジャンル、通知開始条件、通知デバイス) から細かいタスクを生み出す。タスクはそれぞれ使うデバイスが違う。

```
1 (Goal hasGenre Xgenre)
2 (Goal hasTrigger XTrigger)
3 (Goal hasAimObject XAimObject)
4 ->
5 (Goal hasTask XTask1)
6 (XTask1 useDevice XDevice1)
7 (Goal hasTask XTask2)
8 (XTask2 useDevice XDevice2)
9 (Goal hasTask XTask3)
10 (XTask3 useDevice XDevice3)
```

リスト 3.1: タスクの推論タスクの推論 (プレフィックスは省略)

3.5.2 通知と応答の推論

ある通知に対する目標とそれぞれのタスクに使用するデバイスを与えられたら、そこから通知と応答を結びつける推論をする。リスト 3.2 のルールでは、目標の 3 つの要素 (ジャンル、通知開始条件、通知デバイス) 及びタスクに使用するデバイスから通知と応答に関するデバイスの知識を生み出す。

```
1 (Goal hasGenre Xgenre)
2 (Goal hasTrigger XTrigger)
3 (Goal hasAimObject XAimObject)
4 (Goal hasTask XTask1)
5 (XTask1 useDevice XDevice1)
6 (Goal hasTask XTask2)
7 (XTask2 useDevice XDevice2)
8 ->
9 (Goal hasLinker XLinker)
10 (Xlinker hasNotification XNotification1)
11 (XNotification1 control XAimObject)
12 (Xlinker hasResponse XResponse1)
13 (XResponse1 control XDevice1)
14 (XResponse2 control XDevice2)
```

リスト 3.2: 通知と応答の推論 (プレフィックスは省略)

3.6 コンテキストの配信

コンテキストの配信について、応答後の結果を基盤が出版 (publish) し、アプリケーションは API を介して応答成否を購読 (subscribe) する出版-購読型モデルとなる。

第4章 双方向通知応答基盤の実装

本章では双方向通知応答基盤の実装方法について記述する。

4.1 実装の概要

双方向通知応答基盤の開発は Java を使い、知識の概念記述には OWL、個体記述には RDF を用いる。また、実デバイスを直接用いるために Echowand[33]、知識表現をプログラムで扱うためのライブラリ Apache Jena[32] を用いる。

双方向通知応答基盤の大まかな流れは次のようになる。

1. ユーザがアプリケーションを用いて設定した通知のタイプ・通知の開始条件・通知の発生デバイスから推論を行い、タスク及び通知と応答のデバイスを決定する。
2. 通知の開始条件を満たし、通知を開始する。
3. ユーザの行動をセンシングし、通知に対する応答か確かめる。
4. 応答の成否に基づいてアプリケーションに通知成否を返答する。

リスト 2 番目の開始条件については、アプリケーション使用ユーザが屋内外のどこに位置するかを定期的にセンシングしておき、屋内で通知デバイスに近い部屋にいるなら通知デバイス直近のセンサを用いる。一方屋外であったり通知デバイスから離れた部屋であればスマートフォンに通知を行い、通知に対するアクション (通知バーで通知内容をタッチして確認) を応答とする。

4.2 通知と応答の各種パターンの実装

4.2.1 現状認識通知及び動作・状態変化通知

現状認識通知及び動作・状態変化通知は新しいタスクが発生するわけではないので、重要性が低い。そのため、暗黙応答を取得するに留める。暗黙応答は視線移動・身体方向の変更などが考えられるが、本研究では身体の移動のみをセンシングする。例えば、現状認識通知として時計の時報通知があった時は、時計を確認するために接近する場合のみをセンシングする。他に、動作・状態変化通知として給湯器リモコンが操作優先権変更を通知する場合も、暗黙応答として台所パネルへの接近のみをセンシングする。

4.2.2 動作終了通知、呼び出し通知、許容値外通知 a

動作終了通知、呼び出し通知、許容値外通知 a の場合は、ユーザが通知デバイスに接近し新しいタスクを行う明示的な応答をセンシングする。接近したその場で操作を一度行えば新規タスク実行が完了するため、ユーザは暗黙応答の他に通知デバイスに付属するセンサで現場応答を一度行う。例えば動作終了通知として炊飯器のご飯が炊けた通知の場合は、暗黙応答として接近をセンシングし、その後に炊飯器を開けたかどうかを現場応答としてセンシングする。これによってご飯をかき混ぜるというタスクを実行したか判断する。また、呼び出し通知の玄関ドアが開けっ放しの時の通知であれば、暗黙応答判断の後に扉を閉めたかどうかを現場応答としてセンシングする。他に許容値外通知 a の給湯タンクの湯が少ないという通知であれば、暗黙応答判断の後に湯だめボタンを押したかどうかを現場応答とする。

4.2.3 許容値外通知 b、異常通知

許容値外通知 b、異常通知の場合は、暗黙応答、現場応答の後に手続き応答を行う。これら 2 つの通知の場合は、許容値外である状態及び異常な状態を元に戻す復帰応答をすることになる。例えば許容値外通知として部屋温度が上限を超えたことを温度計が通知した場合、暗黙応答の後に窓を開ける明示的な応答を行う。しかし、温度が許容内でなければならぬので、時間が経過し適温になったら復帰応答完了とみなす。他に冷蔵庫の機能が故障した場合、暗黙応答の後に冷蔵庫の故障に気づき電源を切るのが明示的な応答となる。しかし、タスクとしては冷蔵庫の機能の復帰が重要なので、電源を再び付けたときに故障状態が直ってようやく復帰応答完了となる。

4.2.4 ユーザタスク通知

ユーザタスク通知はこれまでのパターンに当てはまらない、複雑な応答をユーザが定義する場合の通知である。例えば目覚まし時計のアラートを考えた時、呼び出し通知でアラームを止める現場応答をしたらタスク完了とすれば良いとする場合と、手続き応答として部屋の外まで出るまでを応答としたいと考える場合がある。後者の場合、ユーザの意図を開発者が予めルールとして記述し、応答取得のためのセンサも用意する必要がある。このようにユーザの目標が設定次第で変わる場合をサービス依存応答として対応する。

4.2.5 緊急通知

緊急通知はユーザやユーザの財産に危険がある場合の通知である。例えば有毒ガスを検知した場合は、通知している場所に行く必要はなく、速やかに家の外に避難するか消防署に電話で連絡すべきである。このように、ユーザタスク通知と同じように複雑な応答をユーザが定義する必要がある。

4.3 双方向通知応答基盤 API の提供

本研究では通知及び応答デバイスを予め準備した状態で、ユーザは API を通じて双方向通知応答基盤を利用する。API の入力は { 通知のタイプ, 通知の開始条件, 通知の発生デバイス } となる。一方 API の出力は { 暗黙応答、明示応答、手続き応答 } のそれぞれの成否 (true か false) となる。

4.4 プログラム上の実装

今回の実装における開発環境は以下の通りである。

- OS : Ubuntu18.04LTS 64bit
- メモリ : 15.4GiB
- プロセッサ : Intel Core i7-3770K CPU @ 3.50GHz × 8
- 開発言語 : Java 11 OpenJDK
- 統合開発環境 : Eclipse Photon Release (4.8.0)

双方向通知応答基盤の全体のクラスの関係は UML を用いたクラス図 4.1 となる。図の赤い部分がデバイス系、黄色の部分がリソース系、青色の部分が推論系・状態遷移系となる。デバイス系とリソース系、リソース系と状態遷移系は Listener クラスや Event クラスを通じて通信を行う。

デバイス系は Echonet Lite 実デバイスの発見・通信や、デバイスオブジェクトの管理を行う部分である。DeviceManager クラスはホームネットワーク内の IP ノードを探し、EchonetLiteNode インスタンスとして管理する。EchonetLiteNode クラスでは Echonet Lite の仕様に従って、IP ノードの先にある実デバイスのオブジェクト情報を特定し、eDataObject のサブクラスをインスタンス化して管理する。eDataObject のサブクラスとして、例えば人間が近づいたかどうか判断する接近センサを表す eHumanDetectionSensor がある。

リソース系は RDF フォーマットに従ったデータの操作を行う部分である。ResourceManager クラスは初期設定や推論によって RDFBaseClass のサブクラスをインスタンス化する。ただし、インスタンス化は RDFClassFactory クラスを通して行う。RDFBaseClass と RDFDevice はアブストラクトクラスであり、インスタンス化するのは最下段のサブクラスとなる。例えば接近センサを RDF 形式で表現する RDFHumanDetectionSensor がある。

推論系は Logic クラスに当たり、ResourceManager クラスを通して推論する機能がある。必要であれば RDFClassFactory クラスを通して新しい情報を RDF 形式で表現する。

状態遷移系はデバイスやユーザの状態の遷移を管理し、それらの状態から通知に対する応答が適切に行われたか判断する。StateManager クラスはある通知要求に対する応答成

否に関わる全ての状態(ユーザ、デバイスなど)を管理する。個別の状態遷移は BaseState-Class のサブクラスをインスタンス化し管理する。例えば、接近センサの状態遷移を管理する SHumanDetectionSensor クラスと、その補助クラスの SHumanDetectionSensorContext クラスがある。

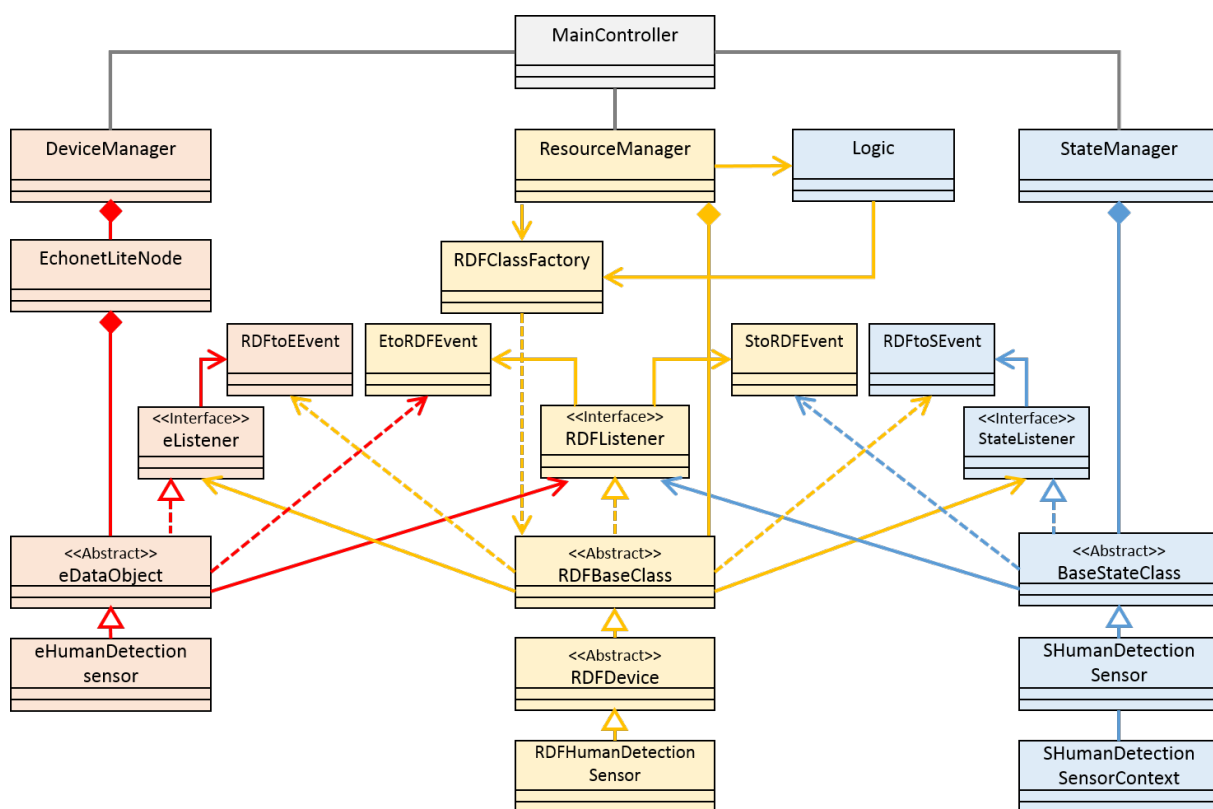


図 4.1: 双方向通知応答基盤の全体のクラス図

eDataObject、RDFBaseClass、BaseStateClass それぞれのサブクラスをインスタンス化したオブジェクト同士は Listener クラス及び Event クラスで通信し合うことができる設計とした。デバイスの状態変更を読み取ったら、デバイス系→リソース系→状態遷移系という順番で情報を送る。また、通知すべきタイミングが状態遷移で発生した時は、状態遷移系→リソース系→デバイス系という順番で情報を送る。これによって、常にリソース系の情報を最新に保つことができる。

なお、実行例は付録 B にて説明する。

第5章 双方向通知応答基盤の評価

本論文では双方向通知応答基盤で取り扱える通知を家電機器から調査し、幅広い家電製品に対して通知及び通知に対する応答が可能であることを示す。なお、ここで取り上げる家電機器はホームネットワークにつながり、通知及び応答の結果が検知可能であるとする。

5.1 評価方法

評価は既存の家電機器から通知機能を調べ、図 2.6 のように通知と応答の対応関係と結びつくか判断する。既存の家電機器は2つの視点から調査する。1つ目は、最新の家電機器製品のラインナップから通知機能を割り出し、通知と応答が対応しているかを判断する。2つ目は、製品安全情報 [34, 35] の電子機器の事故事例において、事故に至るまでの現象を分析する。そして、事故を未然に防止または事故後の事後対応を即座に行うために、事故の発生する可能性のある製品における通知と応答の組み合わせを検討する。

5.2 最新の家電機器製品について

最新の家電機器製品の情報は、家電メーカ各社が公開している製品の取扱説明書から入手した。取扱説明書から通知が可能な機能を抽出し、その応答を検討した結果を A.2 節の表 A.2 に示す。なお、異常通知の引き金となる機械故障についてはほとんどの電子機器で共通して起こるため、検討の対象外とした。

今回調査を行った 23 例の通知について、全て応答を当てはめることが出来た。

5.3 事故事例のある家電機器製品について

門田らの研究 [36] では家電機器の製品事故から発生メカニズムを整理し、潜在的なリスクを明らかにした。製品安全事故発生メカニズムは次の3ステップモデルで表現できる。第1ステップは、製造不良・誤使用・設計構造不良といった深層要因である。第2ステップは絶縁体・部品・トラッキング断線・汚染といった所に不良がある信頼性要因である。第3ステップは個体絶縁破壊・電池欠陥・接触抵抗不良といった直結要因である。こうしたプロセスを辿り、最終現象として発火・発煙が起こる。

今回は3ステップで起こる事故要因について、製品に新しく通知を当てて最終現象が起こる前に事故の前兆を伝えることが可能かを調査する。通知に対する応答も検討をし、A.3節の表 A.3 に示す。表 A.3 の「事故経過と通知のタイミング」の列では、事例から推測できる事故経緯を記述し、その経路のうちで通知できるタイミングを○印で示した。

今回調査を行った74の事故事例に対して、69例について通知及び通知に対する応答を当てはめることが出来た。また、24例については最終現象が起こる前までに通知ができることがわかった。

第6章 双方向通知応答基盤の考察

前章で行った双方向通知応答基盤の評価を踏まえ、考察を行う。以下文章では最新の家電機器製品の調査を A、事故事例から調査した製品の調査を B と呼ぶ。

2.3 節で提案した通知と応答の組み合わせ (図 2.6 を参照) について、A と B からモデル化が適切かを考察する。A について、23 例中 20 例の通知に対する応答は複雑な復帰応答やサービス依存応答ではなく、単純な暗黙応答もしくは現場応答となった。残りの例についても復帰応答に該当し、ユーザが指定しなければならないユーザタスク通知/サービス依存応答の例はない。このように、通常動作する家電製品の通知と応答の組み合わせを限定された種類に単純化出来たと言える。また、B について事故事例の 74 例中 69 例は通知と応答が可能であり、45 例は緊急通知/サービス依存応答であった一方、24 例についてはそれ以外の通知と応答に分類できた。このように、事故を防ぐという面において通知の多様性があるモデルと言える。

第7章 今後の課題

本研究で判明した課題について論ずる。

7.1 暗黙応答への幅広い対応

本研究では人間が暗黙応答としてセンシングが容易なデバイスに近づく場合のみを考えたが、より正確な人間の行動把握には視線や体全体の向きなども考慮する必要がある。

7.2 人間の行動・状態把握への対応

家電機器の通知は家族の不特定の人物に行うが、その通知が不適切になる場合がある。例えば、小さな子供にとって炊飯器は熱い水蒸気が出て危ないと判断した時、子供が炊飯器のそばにいる時に炊き上がりの通知をするのは不適切である。このような複雑な条件に対応するためには、個人を識別し適切なタイミングで通知をするという事前設定が必要である。以上のように、人間の行動を常に把握する仕組みは人間とコンピュータのより自然なやり取りのために必要であるが、本研究では通知が発生してから応答が終わるまでの行動把握に留まる。ホームオートメーションを考える時は、今回の通知というインタフェースの研究を含め、様々な状況下における人間の行動・状態把握のメカニズムが必要である。

第8章 まとめ

本研究では、住環境(を管理するコンピュータ)とユーザが自然な情報のやり取りができる双方向通知応答基盤を提案・実装した。まず、ホームネットワークに関連する技術から通知と応答の結びつきを考案した。次に、実デバイス・周辺環境・ユーザなどを表現したオントロジモデルによるコンテキストから、通知応答の推論が可能な双方向通知応答基盤を提案し、実装方法を示した。そして、双方向通知応答基盤で用いている通知と応答の結びつきのモデルについて、様々な家電機器に当てはまるか調査し、モデルは多様な通知と応答の組み合わせを単純な形で示したものであると結論づけた。

謝辞

丹康雄教授には、普段からの研究指導は元より、学生生活から就職活動に至るまで様々な助言を頂戴いたしました。研究内容に悩んでいた時、「応答とはなにか？」という示唆に富んだ言葉を頂きました。本論文のテーマである通知と応答は、この疑問から議論が進み研究の形となったと言っても過言ではありません。感謝致します。

リム勇仁准教授には、毎回のゼミ発表時に様々な視点から研究を見るようご指導を頂きました。学術文献を数多く読み、自分の説を補強すべきであることに気づくきっかけになりました。感謝致します。

審査員をお引き受けいただいた篠田陽一教授には、研究をより価値のあるものへと昇華するためのご助言を頂きました。感謝致します。

副テーマにおいてご指導をお願いした鶴木祐史教授には、副テーマ指導は元より、私の副テーマに様々な音響効果のある仕組みを導入可能だ、という興味深いアドバイスを頂きました。感謝致します。

丹・リム研究室の徳山さん、牧野さん、Mariosさん、Phamさん、湯村さん、Yuさん、Yangさん、林君、塚越君、金子君、柴田君、Tan君、Dat君、北川さん、洪さん、藤巻君、北村君、中西君、山本君、梶山君、丸谷君、中岡君、Yuan君、Liu君、福嶋君、林さん、辛君、前村君、Jakariaさん、Khun君、Fan君にはお世話になりました。感謝致します。

学業、学生生活を共に歩んできた友人たちに感謝致します。

最後に、学生生活をいつも支えてくれた両親と兄弟に感謝致します。

参考文献

- [1] 福田隆弘, “ホームネットワークにおけるプッシュ型情報のユーザへの提示方法に関する研究,” 2008, 北陸先端科学技術大学院大学 修士論文. [Online]. Available: <http://hdl.handle.net/10119/4337>
- [2] 金井拓哉, “ホームネットワークにおけるプッシュ型情報提示システムの実用化に向けた研究,” 2012, 北陸先端科学技術大学院大学 修士論文. [Online]. Available: <http://hdl.handle.net/10119/10448>
- [3] D. S. McCRICKARD, “A Model for Notification Systems Evaluation — Assessing User Goals for Multitasking Activity,” *ACM Transactions on Computer-Human Interaction*, vol. 10, no. 4, pp. 312–338, 2003.
- [4] A. Mehrotra and M. Musolesi, “Intelligent Notification Systems: A Survey of the State of the Art and Research Challenges,” *arXiv preprint arXiv:1711.10171v2*, 2018.
- [5] L. Schomaker and K. Hartung, “A Taxonomy of Multimodal Interaction in the Human Information Processing System,” *A Report of the Esprit Project 8579 MIAMI WP 1*, 1995.
- [6] J. Rasmussen, “Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-13, no. 3, pp. 257–266, 1983.
- [7] C. D. Wickens, J. G. Hollands, S. Banbury, and R. Parasuraman, *Engineering Psychology & Human Performance*. Psychology Press, 2015.
- [8] R. Parasuraman, “Designing automation for human use : empirical studies and quantitative models,” *Ergonomics*, vol. 43, no. 7, pp. 931–951, 2000.
- [9] H. Limerick, D. Coyle, and J. W. Moore, “The experience of agency in human-computer interactions: a review,” *Frontiers in Human Neuroscience*, vol. 8, pp. 1–10, 2014.
- [10] D. A. Norman, “Conitive Engineering,” *User centered system design*, pp. 31–61, 1986.

- [11] T. B. Sheridan and W. L. Verplank, “Human and computer control of undersea tele-operators.” MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB, Tech. Rep., 1978.
- [12] T. Inagaki, N. Moray, and M. Itoh, “Trust, Self-Confidence and Authority in Human-Machine Systems,” *IFAC Proceedings Volumes*, vol. 31, no. 26, pp. 431–436, 1998.
- [13] 稲垣敏行, “人の認知・判断の特性と限界を考慮した自動走行システムと法制度の設計解説記事 基礎編 9-1 自動化レベル,” 2018. [Online]. Available: <http://www.css.risk.tsukuba.ac.jp/project/pdf/inagaki-kiso-9-1.pdf> (Accessed 2019-01-30).
- [14] R. Parasuraman and B. Toufik, “Theory and design of adaptive automation in aviation systems,” CATHOLIC UNIV OF AMERICA WASHINGTON DC COGNITIVE SCIENCE LAB, Tech. Rep., 1992.
- [15] M. Scerbo W., “Theoretical Perspective on Adaptive Automation,” *Automation and Human Performance: Theory and Applications*, pp. 37–63, 1996.
- [16] T. Inagaki, “Adaptive Automation: Sharing and Trading of Control,” *Handbook of Cognitive Task Design*, pp. 147–169, 2003.
- [17] C. Nass, Y. Moon, B. J. Fogg, B. Reeves, and C. Dryer, “Can Computer Personalities Be Human Personalities?” *ACM*, pp. 228–229, 1995.
- [18] L. D. Turner, S. M. Allen, and R. M. Whitaker, “Reachable but not receptive : Enhancing smartphone interruptibility prediction by modelling the extent of user engagement with notifications,” *Pervasive and Mobile Computing*, vol. 40, pp. 480–494, 2017.
- [19] C. M. Chewar, D. A. Bowman, J. M. Carroll, D. S. Hix, and A. G. Sutcliffe, “User-Centered Critical Parameters for Design Specification , Evaluation , and Reuse : Modeling Goals and Effects of Notification Systems,” Ph.D. dissertation, Virginia Tech, 2005.
- [20] T. Montanaro, “IoT Notifications : from disruption to benefit,” Ph.D. dissertation, Politecnico di Torino, 2018.
- [21] “エコーネット規格（一般公開）ECHONET Lite 規格書 Ver.1.13（日本語版）,” 2018. [Online]. Available: https://echonet.jp/spec_v113_lite/ (Accessed 2019-01-30).
- [22] “エコーネット規格（一般公開）APPENDIX ECHONET 機器オブジェクト詳細規定 Release K,” 2018. [Online]. Available: <https://echonet.jp/wp/wp-content/uploads>

- /pdf/General/Standard/Release/Release_K.jp/Appendix_Release_K.pdf (Accessed 2019-1-30).
- [23] “Published Drafts TS-0012 oneM2M Base Ontology,” 2018. [Online]. Available: http://onem2m.org/component/rsfiles/download-file/files?path=Release_3_Draft_TS%255CTS-0012-Base-Ontology-V3_7_1.docx&Itemid=238 (Accessed 2019-2-2).
 - [24] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context aware computing for the internet of things: A survey,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.
 - [25] H. Chen, T. Finin, and A. Joshi, “An ontology for context-aware pervasive computing environments,” *Distributed Systems, Knowledge Engineering Review*, vol. 18, no. 3, pp. 197–207, 2004.
 - [26] R. Hervas and J. Bravo, “COIVA : context-aware and ontology-powered information visualization architecture,” *Software: Practice and Experience*, vol. 4, no. 41, pp. 403–426, 2011.
 - [27] V. C. PHAM, Y. Lim, A. SGORBISSA, and Y. TAN, “An Ontology-Driven ECHONET Lite Adaptation Layer for Smart Homes,” *Journal of Information Processing(JIP), Special issues of Information System*, vol. 27, no. 3, pp. 1–9, 2019.
 - [28] V. C. PHAM, “GitHub - Cupham/EchonetOntology: Java Code of Echonet Ontology.” [Online]. Available: <https://github.com/Cupham/EchonetOntology> (Accessed 2019-02-09).
 - [29] J. Swetina, “The oneM2M Base Ontology.” [Online]. Available: <https://git.onem2m.org/MAS/BaseOntology> (Accessed 2019-2-14).
 - [30] V. Link, S. Lohmann, E. Marbach, S. Negru, and V. Wiens, “WebVOWL: Web-based Visualization of Ontologies.” [Online]. Available: <http://vowl.visualdataweb.org/webvowl.html> (Accessed 2019-02-04).
 - [31] M. Lefrançois and J. Kalaoja, “Smart Energy Aware Systems The SEAS Building Ontology,” 2016. [Online]. Available: <https://ci.mines-stetienne.fr/seas/BuildingOntology-1.0> (Accessed 2019-2-14).
 - [32] “Apache Jena, A free and open source Java framework for building Semantic Web and Linked Data applications.” [Online]. Available: <https://jena.apache.org/> (Accessed 2019-2-14).

- [33] M. Yoshiki, “echowand - Yet Another ECHONET Lite Library for Java.” [Online]. Available: <https://github.com/ymakino/echowand> (Accessed 2019-2-14).
- [34] “製品安全ガイド, (METI/経済産業省).” [Online]. Available: http://www.meti.go.jp/product_safety/index.html (Accessed 2019-02-14).
- [35] “Safety Publications, U.S. Consumer Product Safety Commission (CPSC).” [Online]. Available: <https://www.cpsc.gov/Safety-Education/Safety-Guides/General-Information/Publications-Listing> (Accessed 2019-02-14).
- [36] 門田靖 and 田中健次, “潜在リスク抽出のための事故情報解析技法の提案,” **日本信頼性学会誌 信頼性**, vol. 38, no. 1, pp. 57–66, 2016.

付 録 A 製品調査

A.1 Echonet Lite 機器および oneM2M 機器に基づく家電機器の既存製品調査

表 A.1: Echonet Lite 機器および oneM2M 機器に基づく家電機器の既存製品調査

ID	クラス	メーカ	製品名	通知場所	通 知 ト リ ガー	通知方法	通知デバイス	通知	応答	応答デバイス
1	電 気 温 水 器 ク ラ ス 0x6B、瞬 間 式 給 湯 器 ク ラ ス 0x72	三菱	ヒートポン プ給湯器	台所・浴室	「優先」ボタ ンを押す	報知音、音声 ガイダンス	浴室パネル 台所パネル	動作変化	暗黙	位置センサ
2					給湯温度を 60 度にする	報知音、音声 ガイダンス	浴室パネル 台所パネル	許容値外	現場	位置センサ ボタン
3					タンクのお 湯が少なく なる	報知音	浴室パネル 台所パネル	許容値外	現場	位置センサ ボタン
4					一日の蛇口 /シャワー使 用量が目標 湯量を超える	報知音・アナ ウンス	浴室パネル 台所パネル	タスク	サービス依 存:使用量を 今後減らす	位置センサ ボタン 内部 センサ
5					「通話」ボタ ンを押す	呼出音・ラン プ	浴室パネル 台所パネル	呼び出し	現場	位置センサ ボタン マイ ク
6	オープンレ ンジクラス 0xB8	東芝	過熱水蒸気 オープンレ ンジ	居間	過熱終了	ブザー 3 回	レンジ	動作終了	現場	位置センサ 扉センサ ボ タン
7					トーストを 裏返す時	ブザー 5 回	レンジ	呼び出し	現場	位置センサ 扉センサ ボ タン
8					余熱終了	ブザー 5 回	レンジ	呼び出し	現場	位置センサ 扉センサ ボ タン
9					異常発生	ブザー 7 回	レンジ	異常	復帰	位 置 セ ン サ 電 圧 セ ン サ (電 源 OFF) 電 話
10					水補給が必要 な時	ブザー 7 回	レンジ	許容値外	復帰	位置センサ 水位センサ
11					扉がしっかり 閉まってい ない時	ブザー 5 回	レンジ	呼び出し	現場	位置センサ 扉センサ ボ タン
12	家庭用エア コンクラス 0x30	ダイキン	ルームエア コン	居間 寝室 な ど	冷房運転中 に屋外温度 が設定温度 より低くな った時	音声ガイダ ンス	エアコン	タスク	サービス依 存:窓を開け る	位置センサ 窓開閉セン サ ボタン (エアコン) 温度センサ
13					快適自動運 転中に設定 を変更後、 30 分・1 時 間経つ	音声ガイダ ンス	エアコン	動作・状態変 化	暗黙	位置センサ
14					留守エコ設 定で設定時 刻になり、か つ人がいな い時	音声ガイダ ンス	エアコン	動作・状態変 化	応答必要な し	-
15	電気錠クラ ス 0x6F	Panasonic	電気錠操作 ユニット	リビング	ドアを開け た時	開戸報知音 が 2 回	操 作 ユ ニ ッ ト	動作・状態変 化	暗黙	位置センサ

table continued on next page

ID	クラス	メーカー	製品名	通知場所	通知トリガー	通知方法	通知デバイス	通知	応答	応答デバイス
16					ドアをこじ開けた時、ドアとドア枠の隙間が大きい時、警戒モードでサムターンやキーで開けた時	警報音(ピーポー)が5分間	操作ユニット	緊急	サービス依存:緊急行動	位置センサ 扉センサ ボタン 電話
17	ブザークラス 0xA0 alarmSpeaker[tone=2] (theft 防犯)	Secual	スマート防犯窓センサー	窓	窓センサーが衝撃を検知	警報音	ゲートウェイ	緊急	サービス依存:緊急行動	位置センサ ボタン 電話
18	冷凍冷蔵庫クラス 0xB7, doorStatus, deviceRefrigerator	Samsung	Family Hub Multi-door Fridge Freezer	台所	ドアが開けばなし	ピープ音	冷蔵庫	呼び出し	現場	位置センサ 扉センサ
19					冷蔵庫、冷凍室が異常な温度の時	スマホでアラーム	スマートフォン (冷蔵庫)	異常	復帰	位置センサ 電圧センサ (電源OFF) 電話
20					一定期間が過ぎ、水フィルターの交換が必要	スマホでアラーム	スマートフォン (冷蔵庫)	許容値外	復帰	位置センサ 交換時期のリセット操作
21	炊飯器クラス 0xBB	Panasonic	スチーム IH ジャー炊飯器	台所	炊き上がり時	ブザーが鳴る	炊飯器	動作終了	現場	位置センサ 開閉センサ ボタン
22					スマホアプリで炊き上がり時間予約をしておき、予約時間に近づく	5分前にスマホで通知	スマートフォン	現状認識	暗黙	位置センサ
23	洗濯機クラス 0xC5、 衣類乾燥機クラス 0xC6、 洗濯乾燥機クラス 0xD3	Panasonic	ドラム式電気洗濯乾燥機	脱衣所	洗濯・乾燥終了	ブザーが鳴る	洗濯乾燥機	動作終了	現場	位置センサ 開閉センサ ボタン
24					花粉ケア・除菌消臭コース終了	ブザーが鳴る (しわが出来るのですぐに取り出す)	洗濯乾燥機	動作終了	現場	位置センサ 開閉センサ ボタン
25					槽乾燥、槽洗浄終了	ブザーが鳴る	洗濯乾燥機	動作終了	現場	位置センサ ボタン
26					エラー時	ブザーが鳴る	洗濯乾燥機	異常	復帰	位置センサ 電圧センサ 電話
27					自動投入用の洗剤が少ないとき	スマホアプリで通知	スマートフォン (洗濯乾燥機)	許容値外	復帰	位置センサ 開閉センサ ボタン
28	alarmSpeaker[volume=1] (Fire 火災報知) smoke-Sensor	Honeywell	Smoke detector	台所 廊下 寝室 など	煙の検知時	本体と専用ゲートウェイのアラームが鳴る	火災報知器ゲートウェイ	緊急	サービス依存:緊急対応	位置センサ 扉センサ 電話センサ ボタン
29					故障時	ピープ音が鳴る	火災報知器	異常	復帰	位置センサ 電圧センサ 電話
30					電池容量が低下した時	ピープ音が鳴る	火災報知器	許容値外	復帰	位置センサ 電圧センサ ボタン
31	TemperatureAlarm	Ardo	アラーム付き温湿度計	リビング ダイニング 寝室 など	温度・湿度が上限値・下限値を超えた時	ブザー音	温湿度計	許容値外	復帰	位置センサ 開閉センサ 温度センサ ボタン
32	waterSensor	ZIRCON	Leak Alert WiFi	脱衣所	水漏れの検知時	アラーム音	水漏れ検知器	許容値外	復帰	位置センサ 水反応センサ ボタン
33	alarmSpeaker[volume=1] (doorbell ドアベル)	Panasonic	SpotCam-Ring	玄関	本体ボタンが押される	チャイム スピーカーから音が鳴る	ドアホン	呼び出し	現場	位置センサ ボタン

table continued on next page

ID	クラス	メーカー	製品名	通知場所	通知トリガー	通知方法	通知デバイス	通知	応答	応答デバイス
34					本体の電池が少ない、電池カバーが外された、本体がオフラインになった時	メールを送る	スマートフォン (ドアホン)	許容値外	復帰	位置センサ ボタン 通信状態
35	phoneCall	Panasonic	コードレス電話機	リビング	電話がかかる	親機、子機が鳴る	電話	呼び出し	現場	位置センサ 電話 ボタン
36					留守番セットしているときに電話がかかる	呼び出し音が 4 回鳴る	電話	呼び出し	現場	位置センサ ボタン 電話
37					ドアホンの呼び出しが押される	呼び出し音	電話	呼び出し	現場	位置センサ ボタン 電話
38					窓・ドアセンサーが開閉を検知	侵入者に対する警報音	電話 スマートフォン	緊急	サービス依存:緊急対応	位置センサ ボタン
39					人感センサー付子機が人を検知	携帯に電話	スマートフォン	緊急	サービス依存:緊急対応	電話
40	電動ブラインド・日よけクラス 0x60	Brunt	Blind Engine	リビング 寝室 など	タイマーの時間になる	シャッターが開く	ブラインド	動作・状態変化	暗黙	位置センサ
41	電動シャッタークラス 0x61 電動雨戸・シャッタークラス 0x63	文化 シャッター	セレクトネット	リビング 寝室 など	お日様タイマー設定・お好みタイマー設定 (全開・半開・換気・採光・角度 1-6) をし、時間になる	設定どおりの開閉状態になる	シャッター	動作・状態変化	暗黙	位置センサ
42					シャッターの状態が変化した時	メールを送る	スマートフォン	動作・状態変化	暗黙	受信端末操作
43					定期的にシャッター状態を通知する	メールを送る	スマートフォン	現状認識	暗黙	受信端末操作
44	電動窓クラス 0x65	VEKTIVA	smarwi	リビング 寝室 など	タイマーの時間になる	窓が開く	窓	動作・状態変化	暗黙	位置センサ
45	一般照明クラス 0x90 など	Philips	Hue	リビング 寝室 など	スケジュールを設定し、時間になる	設定どおりの照明	照明	タスク	サービス依存:設定どおりの応答	ex) 目覚まし 位置センサ
46					リマインドタイマーを設定し、時間になる	設定どおりの照明	照明	タスク	サービス依存:設定どおりの応答	ex) 勉強の時間 位置センサ 照明の電圧センサ

A.2 最新の家電機器製品の調査

表 A.2: 最新の家電機器製品における通知と応答の調査

id	メーカー	デバイス名	通知トリガ	通知方法	通知デバイス	通知種	応答	応答デバイス	url
1	Sharp	液晶カラーテレビ LC-70X500	おはようタイマーの設定時間になる	電源がつき、画面が点灯しアラームが鳴る	TV	呼び出し	現場	位置センサリモコンボタン	http://www.sharp.co.jp/support/aquos/doc/1c70x500_mn.pdf?productId=LC-70X500
2	Sharp	4K レコーダー 4BC20AT3	通知センターによる本機器及び放送局からの通知情報が発生 (HDD 残量が少ない・ソフトウェアの更新・BD ドライブのお手入れ必要)	画面左下にメッセージがポップする	ディスプレイ	許容値外	復帰	位置センサリモコンボタン 本体	http://www.sharp.co.jp/support/av/dvd/doc/4bc40_20at3_mn.pdf?productId=4B-C20AT3
3	Sharp	ファクシミリ UX-AF91CL	電話がかかってくる	親機、子機が鳴る	ファクシミリ	呼び出し	現場	位置センサボタン ファクシミリ	http://www.sharp.co.jp/support/fax/station/doc/uxaf91_mn.pdf?productId=UX-AF91CL
4	Sharp	ファクシミリ UX-AF91CL	留守番セット時に電話がかかる	親機から設定した回数に着信音、応答メッセージ、ピーという発信音が鳴る	ファクシミリ	呼び出し	現場	位置センサボタン ファクシミリ	http://www.sharp.co.jp/support/fax/station/doc/uxaf91_mn.pdf?productId=UX-AF91CL
5	Sharp	ファクシミリ UX-AF91CL	ドアホンが押される	親機、子機が鳴る	ファクシミリ	呼び出し	現場	位置センサボタン ファクシミリ	http://www.sharp.co.jp/support/fax/station/doc/uxaf91_mn.pdf?productId=UX-AF91CL
6	Sharp	空気清浄機 KIJP100	空気清浄やフィルタ掃除完了後	音声で完了したことを伝える	空気清浄機	動作・状態変化	暗黙	位置センサ	http://www.sharp.co.jp/support/air_purifier/doc/kijp100_mn.pdf?productId=KI-JP100-W
7	Sharp	空気清浄機 KIJP100	見張り機能中に高温・高温または低温・感想になった時	音声で危険を知らせて運転を切り替える	空気清浄機	動作・状態変化	暗黙	位置センサ	http://www.sharp.co.jp/support/air_purifier/doc/kijp100_mn.pdf?productId=KI-JP100-W
8	Sharp	洗濯機 ESU111	洗濯物が残り 3 分で取り出せる	音声で残り時間を伝える	洗濯機	現状認識	暗黙	位置センサ	http://www.sharp.co.jp/support/washer/doc/esu111_mn.pdf?productId=ES-U111#page=1&zoom=auto,594,879
9	Sharp	洗濯機 ESU111	洗濯・乾燥・ドラム清掃が終わる	音声で終了を伝える	洗濯機	動作終了	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/washer/doc/esu111_mn.pdf?productId=ES-U111#page=1&zoom=auto,594,879
10	Sharp	冷蔵庫 SJGX50E	ドアを開け忘れて 1 分以上経つ	ブザーが鳴る	冷蔵庫	呼び出し	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/refrigerator/doc/sjgx50e_55e_mn.pdf?productId=SJ-GX50E
11	Sharp	冷蔵庫 SJGX50E	タイマー冷凍を設定し、30 分間冷やした後	ブザーが鳴る	冷蔵庫	動作終了	現場	位置センサタッチパネル 開閉センサ	http://www.sharp.co.jp/support/refrigerator/doc/sjgx50e_55e_mn.pdf?productId=SJ-GX50E
12	Sharp	ウォーターオープン/電子レンジ AXW500	加熱が終了した	音声で終了を伝える	電子レンジ	動作終了	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/healsio/doc/axw500_mn.pdf?productId=AX-XW500
13	Sharp	ウォーターオープン/電子レンジ AXW500	予熱が終了した	音声で予熱終了を伝える	電子レンジ	呼び出し	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/healsio/doc/axw500_mn.pdf?productId=AX-XW500
14	Sharp	ホット クック KNHW24C	加熱が終了した	音声で終了を伝える	ホット クック	動作終了	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/hotcook/doc/knhw24c_mn.pdf?productId=kn-hw24c
15	Sharp	ホット クック KNHW24C	加熱を一時中断し、食材を追加または食材を取り出す必要がある時	音声で指示を伝える	ホット クック	呼び出し	現場	位置センサ開閉センサ	http://www.sharp.co.jp/support/hotcook/doc/knhw24c_mn.pdf?productId=kn-hw24c

table continued on next page

id	メーカー	デバイス名	通知トリガ	通知方法	通知デバイス	通知種	応答	応答デバイス	url
16	Sharp	ホット クック KNHW24C	操作の最後にスタート キーを押すのを忘れ、時 間が経つ	ブザーが鳴る	ホット クック	呼び出し	現場	位置センサ ボタン	http://www.sharp.co.jp/support/hotcook/doc/knhw24c_mn.pdf?productId=kn-hw24c
17	Sharp	お茶メーカー TE-TS56V	茶葉を挽き終わった	ブザーが鳴る	お茶 メーカー	動作終了	現場	位置センサ 接触センサ	http://www.sharp.co.jp/support/ocha/doc/tets56v_mn.pdf?productId=TE-TS56V
18	Sharp	お茶メーカー TE-TS56V	茶を沸かし終わった	ブザーが鳴る	お茶 メーカー	動作終了	現場	位置センサ 接触センサ ボタン	http://www.sharp.co.jp/support/ocha/doc/tets56v_mn.pdf?productId=TE-TS56V
19	Sharp	ジャー炊飯器 KS-CF05A	ご飯が炊き終わった	ブザーが鳴る	ジャー 炊飯器	動作終了	現場	位置センサ 開閉センサ	http://www.sharp.co.jp/support/ricecooker/doc/kscf05a_mn.pdf?productId=KS-CF05A
20	Sharp	ペットケア モニター HNPC001	ペットの調子が悪いこと がわかった時	アラート通知を行う	スマー トフォ ン	異常	復帰	タップ位置 センサ	http://www.sharp.co.jp/support/pethhealthcare/doc/hn-pc001_mn.pdf?productId=HN-PC001
21	Sharp	布団乾燥機 UD-AF1	布団などの乾燥・あたため・ダニ対策・消臭が 終わった	ブザーが鳴る	布団乾 燥機	動作終了	現場	位置センサ 開閉センサ	http://www.sharp.co.jp/support/kansouki/doc/udaf1_mn.pdf?productId=UD-AF1
22	Sharp	加湿器 HVVH75	給水が必要な時	ブザーが鳴り、給水マー クが点滅する	加湿器	許容値外	復帰	位置センサ 接触センサ 水位センサ	http://www.sharp.co.jp/support/air_purifier/doc/hvh55_75_mn.pdf?productId=HV-H75
23	Sharp	加湿器 HVVH75	加湿時間の累計が 240 時間を越す	ブザーが鳴り、お手入れ ランプが点灯する	加湿器	許容値外	復帰	位置センサ 接触センサ	http://www.sharp.co.jp/support/air_purifier/doc/hvh55_75_mn.pdf?productId=HV-H75

A.3 事事故事例のある家電機器製品の調査

表 A.3: 事事故事例から調査した製品における通知と応答

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
1	ヘヤードライヤーの本体出口部の電源コードが焦げて電源が入らない。	ドライヤー	電源コード付け根部分の損傷によるショート	目視確認	経時劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/code.pdf
2	電気あんかを足下に置いて就眠したが、足に熱を感じて取りだしてみると火花が出ており、コードの根本が焦げていた。	電気あんか	電源コード付け根部分の損傷によるショート	目視確認	経時劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/code.pdf
3	寝る前の布団を予め暖めるために電気毛布のスイッチを入れておいたところ、しばらくして炎が上がった。コントロール付け根のコードが断線していた。	電気毛布	電源コード付け根部分の損傷によるショート	目視確認	経時劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/code.pdf
4	洗濯が終了したと思い、上ぶたを開けると脱水槽がゆっくり回っていたが、洗濯物を取ろうと手を入れたところ、洗濯物が絡まり、指を切断した。	洗濯機	ロック機構が破損	緩い回転であっても手を入れてはいけませんロック機構が故障していたり、脱水槽のブレーキの効きが悪くなっている場合は、直ちに使用を中止	設計・構造不良&○誤使用→事故	回転中にふたを開けた	異常通知/復帰応答	N:ブザー R1: [近づく] R2:ふたを閉じる R3:電源を切る	http://www.meti.go.jp/product_safety/consumer/pdf/sentakuki.pdf
5	定格容量1500Wのマルチタップに700Wのホットカーペットと1000Wの電気ポットを同時に使用していたところマルチタップのコード部から出火した。	マルチタップ	マルチタップに過度の電流が通電し、発生した熱により配線被覆が溶融し、マルチタップの配線内で短絡	マルチタップに接続されるそれぞれの電気製品の消費電力に関する表示を確認し、合計する消費電力がマルチタップの定格容量を超えないことを確認	設計・構造不良→○過負荷→発火	電流量が制限値を超えた	許容値外通知/現場応答	N:ブザー R1:近づく R2:電流量を0にする(電源を切る)	http://www.meti.go.jp/product_safety/consumer/pdf/tup.pdf
6	リビングルーム壁際に置いた観賞魚用水槽上部から発火した。	水槽ライト	水槽上部にガラスフタが無く、飼育水の飛沫がライトにかかり、水槽ライトに付属の使用中的サービスコンセントに水が溜まったことにより、コンセントでトラッキング現象を起こして発火した	水槽の近くに電気機器が置いてある水槽にあっては、フタを取り付けるなどして飛沫が飛び散るのを防ぎましょう。製品の電源コンセントは飛沫がかからず、ちり、ほこりがたまらない場所に設置しましょう。	○誤使用→トラッキング→発火	ふたが一定時間開けっ放しであることを検知する	許容値外通知/現場応答	N:ブザー R1:近づく R2:ふたを閉じる	http://www.meti.go.jp/product_safety/consumer/pdf/toraking.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
7	開放式小型ガス瞬間湯沸器から黒煙が出ているのを以前から知りつつも点検・修理をせずに使用を継続していたところ、目眩がして気分が悪くなり、病院に運ばれCO中毒と診断された。	ガス瞬間湯沸器	長年におたる使用で熱交換器が汚損して目詰まりを生じていたことから、不完全燃焼となり黒煙が上がると共に一酸化炭素が発生していた。	黒煙やススが発生しているときや、湯沸器上部周囲がススで黒く汚れているときは、不完全燃焼となっている状態のため高濃度の一酸化炭素が発生し危険ですから、直ちに使用を中止し、業者に点検を依頼しましょう。	誤使用&経時劣化&保守・設置不良→○CO発生	COを検知する	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/k-yuto_hurogama.pdf
8	ベランダに設置した屋外設置型ガス給湯器付近から発火し、ベランダの一部を焼損した。	屋外ガス給湯器	排気口から可燃物までの距離が近いと、可燃物の発火に至ることがあります。	排気口近くに洗濯物を干したり、可燃物を置いてはいけません。給湯器の周りを段ボール箱や板などで囲ってしまうと給気不足となり異常燃焼を起こし高温ガスが排気口から溢れ出す危険性があります	○誤使用→発火	物体検知センサが長時間置かれた物体を検知	許容値外通知/現場応答	N:ブザー R1:近づく R2:物体を検知しなくなる	http://www.meti.go.jp/product_safety/consumer/pdf/k-yuto_hurogama.pdf
9	浴室で家人がCO中毒により死亡した。本来屋内で使用するべきではない屋外設置型のガス給湯器を増設された脱衣所とユニットバスを波板で囲った中に設置していた。	屋外ガス給湯器	屋外設置型のガス給湯器を屋内で使用したことにより排気ガスに含まれる一酸化炭素が拡散されずに危険な濃度にまで高まって建家内に流入した。	燃焼のための吸排気が正常にできず不完全燃焼から一酸化炭素中毒を発生するおそれがありますから、屋外設置型のガス給湯機の周囲を波板やビニールシート等で囲ってはいけません。	誤使用→○CO発生	COを検知する	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/k-yuto_hurogama.pdf
10	石油ふろがまのバーナーに点火してしばらくしたら、バーナー全体から火が出たので消火器で消火した。このところ油の臭いが気になっていた。	石油ふろがま	灯油タンクとバーナーとの間をつなぐゴム製送油管に亀裂が生じたため、送油管から灯油が漏れ、何らかの火が引火した。ゴム製送油管は、経年劣化や無理な折り曲げなどにより亀裂が生じることがあります。	送油管に亀裂が入っていないかどうか、日頃より目視点検し、もし亀裂が見つかった場合には、速やかに販売店など専門家に修理を依頼しましょう。送油管を無理に折り曲げて配管したり、踏みつけたりしてはいけません。設置状態に注意しましょう。	経時劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/k-yuto_hurogama.pdf
11	浴室内に設置されたガスふろがまを追い炊きしているときに、洗濯機の排水を浴室に流していたら、ふろがまの前面等が焼損した。	浴室内ガスふろがま	浴室床面に流れ込んだ水で、ガスバーナーのノズル部が冠水してしまったことにより、ガスがバーナーの外で燃焼し機器の焼損に至った。	風呂がまが冠水すると火災・故障の原因になります。浴室の排水口は、こまめに掃除をして水はけをよくしておきましょう。	保守・設置不良&○異物・汚染・吸水→発火	水量レベルを検知	許容値外通知/現場応答	N:ブザー R1:近づく R2:水量が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/k-yuto_hurogama.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
12	ガスストーブとゴム管の接続部付近からガスが漏洩し、ガスストーブの火に引火して火災になった。当該ガスストーブ本体とゴム管との接続にはソケットを用いることが指定されていたところ、ソケットが無いゴム管を用いて接続していた。	ガスゴム管の接続	当該製品の場合、ソケットを用いた接続とすることが指定されていたところに誤ってソケットが無いホースを接続したため接合部からガスが漏洩した。	ガス栓、ゴム管、ガス機器の間の接続は、接続形状、サイズが適合する接続具が付いたゴム管やガスコードを使用してください。	保守・設置不良&誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kyuto_hurogama.pdf
13	ガス栓を開く際、暗かったため、ライターの火を照明替わりにして、さらにガス栓のツマミが破損していたことから、ペンチで開閉を行っていた。ガス栓を開いた際、漏れたガスにライターの火が引火し、火災が発生した。	ガス栓	ガス栓をガス栓ツマミがない状態で使用し、ペンチで無理に開けようとしたこと、照明の代わりにライターを用いたため漏れたガスに着火した。	ガス栓のツマミ等が破損、紛失していた状態で使用してはいけません。業者に修理を依頼しましょう。ペンチ等の工具を使ってガス栓を無理に開けてはいけません。	誤使用&保守・設置不良→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kyuto_hurogama.pdf
14	部屋の外に出たくない!石油ストーブの近くで給油。こぼれた灯油に引火し、火災に	石油ストーブ	カートリッジタンクが倒れたり、給油ホースが抜けたらして、こぼれた灯油に引火し、火災につながることがあります。	火の気があるところでの給油はやめましょう。	○誤使用→発火	石油ストーブとタンクが近いことを検知	許容値外通知/現場応答	N:ブザー R1: 近づく R2: ストーブとタンクの距離を取る	http://www.meti.go.jp/product_safety/consumer/pdf/satovepnf.pdf
15	何回も給油するのは面倒だし、ぎりぎりまでいれちゃおう!あふれた灯油に引火すると火災につなが	石油ストーブ	あふれた灯油に引火する	カートリッジタンクの外側やフタに灯油が付いてしまったときは、しっかりふき取りましょう。	○誤使用→発火	給油量のレベルを検知	許容値外通知/現場応答	N:ブザー R1: 近づく R2: 給油タンクからボンブを抜く	http://www.meti.go.jp/product_safety/consumer/pdf/satovepnf.pdf
16	カートリッジタンクのフタは手が汚れるからさわりたくない!灯油がこぼれて、火災につながる	石油ストーブ	フタをきちんと閉めないで、カートリッジタンクをセットするとき、点火するとき、取外すときに、灯油がこぼれて、火災につながることがあります。	石油ストーブのそばに持つていく前に、カートリッジタンクを逆さにして灯油の漏れがないか確認しましょう。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/satovepnf.pdf
17	外に干しても乾かないし冷えるから、ストーブの近くで洗濯物を干すと、落ちて火災	石油ストーブ	ストーブの近くで洗濯物を干すと、落ちて火災の原因となります。炎に直接触れていなくても、燃えるものがストーブの近くにあると、発火して火災につながることがあります。	ストーブの近くで、ものを乾かしたり、暖めたりすることはありませんか?危険なのでやめましょう。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/satovepnf.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
18	ヘアスプレー／防水スプレー／静電気除去スプレー／殺虫スプレー／カセットガスコンロのボンベをストーブの近くで使い、引火	石油ストーブ	スプレー缶には可燃性のガスが含まれている、近くに置くと中のガスが膨張して、爆発することもある	ストーブの近くでは使用はやめましょう	○誤使用→発火	スプレー缶とストーブが近いことを検知	許容値外通知/現場応答	N: プザー R1: 近づく R2: スプレー缶とストーブの距離を取る	http://www.meti.go.jp/product_safety/cons umer/pdf/s tovepnf.pdf
19	衣類がストーブに触れて、溶けたり引火する	石油ストーブ	寒いときには、たくさん着込んであることがあり、気付かない	厚手の衣類や、毛足の長いものには注意しましょう。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N: プザー R1: 離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/s tovepnf.pdf
20	あったかくてばーっとしてきた?一酸化炭素中毒で死亡に至る	石油ストーブ	換気をしていない	窓を開けるなど十分に換気をしましょう。	誤使用→○CO発生	COを検知する	緊急通知/サービス依存応答	N: プザー R1: 離れる R2: 換気する R3: 濃度が下がる	http://www.meti.go.jp/product_safety/cons umer/pdf/s tovepnf.pdf
21	間違えてガソリンを給油して火災、住宅から出火して全焼し、顔や手にやけどを負った	石油ストーブ	ガソリンを灯油と間違えて石油ストーブに給油したため、異常燃焼を起こして火災に至ったものです。	保管場所に注意しましょう。	○誤使用→発火	ガソリン缶とストーブが近いことを検知	許容値外通知/現場応答	N: プザー R1: 近づく R2: ガソリン缶とストーブの距離を取る	http://www.meti.go.jp/product_safety/event/hu yubanoseih injiko.pdf
22	ガスファンヒーターの電源スイッチを入れたら火がつき、機器が焼けた。	ガスファンヒーター	専用のガスコードではなく、ガス用ゴム管を接続したため接続口からガスがもれ、点火した際に引火したものです。	ガス栓とガス機器の接続は適正な接続具を使用してください。使っていないガス栓は専用のガス栓キャップをかぶせてください。接続は赤い線まで差し込んでゴム管止めで抜けないようにしてください。ソケットにごみなどがはさまっていないことを確認してください。	誤使用&保守・設置不良→○発火	発煙を検知	緊急通知/サービス依存応答	N: プザー R1: 離れる	http://www.meti.go.jp/product_safety/event/hu yubanoseih injiko.pdf
23	電気ストーブにふとんがふれて火災、住宅から出火して全焼し、1人が死亡した。	電気ストーブ	電気ストーブをつけたまま就寝したため、ふとんがヒーターに触れて火がつき、出火したものです。	寝るときは必ず電源スイッチを切りましょう。また、そばにカーテンなどの燃えやすいものを置かないでください。洗濯物を乾かしたことが原因による火災事故も多く発生しています。乾燥して軽くなった洗濯物が上昇気流で外れてストーブに落下する危険性があります。	○誤使用→発火	室内ライトが消えた・薄暗くなった時にストーブが点いている	許容値外通知/現場応答	N: プザー R1: ストーブに近づく R2: ストーブを切る	http://www.meti.go.jp/product_safety/event/hu yubanoseih injiko.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
24	カセットコンロで調理中、コンロのカセットボンベが爆発して、2人がやけどを負った。	カセットボンベ	五徳を収納したまま使用していたため、フライパンの底とコンロが接触状態になって加熱され、コンロ全体が過熱してボンベが破裂したものです	カセットボンベを装着する際は、正しく装着してください。五徳が裏返しになっていないかを確認してください。カセットコンロは2台並べて使用しないでください。また、コンロを覆うような大きな鍋などは使用しないでください。	○誤使用→爆発	五徳が正しくセットされないうで火をつける	許容値外通知/ 現場応答	N:ブザー R1: [近づく] R2: 五徳を正しくセットする	http://www.meti.go.jp/product_safety/event/huyubanoseihinjiko.pdf
25	電子レンジ加熱式のゆたんぼを加熱していたら、ゆたんぼが膨らんできたためあわてて電子レンジの扉を開けると、ゆたんぼが破裂して内容物が飛び散り、顔面にやけどを負った。(	電子レンジ加熱式ゆたんぼ	オート加熱機能禁止する表示がありましたが、誤って使用したため、規定時間を超えて加熱され、袋の内圧が高まって破損し、内容物が飛び散ったものです。	取り扱い表示の使用方法及び加熱時間を必ず守ってください。袋が膨張、あるいは内容物が漏れ出た場合は、電子レンジのスイッチを切り、十分に時間をおいて冷めたことを確認してから扉を開けてください。ガスコンロや電磁調理器で直接加熱するタイプの金属製ゆたんぼは、必ず口金（キャップ）を外してから、加熱してください。内圧が上昇して破裂します	誤使用→破裂	×			http://www.meti.go.jp/product_safety/event/huyubanoseihinjiko.pdf
26	ゆたんぼで低温やけどを負った。	ゆたんぼ	長時間、ゆたんぼを使っていたため低温やけどに至ったものです	「低温やけど」は、ゆたんぼやこたつなどのほか、使いすてのカイロなどでも発生します。同じ部位を長時間温めないでください。また、違和感や熱いと感じたら直ちに使用を中止してください。厚手のタオルや専用のカバーなどで包んでも低温やけどを負うことがあります。ゆたんぼは就寝前にふとんの中に入れて、温まったら出して、電気あんかはスイッチを切ってください。	誤使用→やけど	×			http://www.meti.go.jp/product_safety/event/huyubanoseihinjiko.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
27	・浴室内で虫を退治しようとして連続噴射後、入浴のために内釜に点火したところ引火して火災となり、やけどを負い入院。・使用中のコンロ近辺に出現した虫を退治しようとして噴射したところ引火して火傷を負った	エアゾールスプレー	スプレーには噴射圧力をかけるために可燃ガス（LPガス）が用いられているため、噴射により滞留した可燃ガスに、ガス器具の火が引火した	炎や火気の近く、高温となる場所ではスプレーを使用しない。（表示をよく読む）使用中並びに使用後は十分に換気を行う。	○誤使用→発火	可燃ガスを検知	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/aerosol.pdf
28	シュレッダーで紙詰まりが発生したので、潤滑油をカッター部に供給して回転をスムーズにしようとして潤滑スプレーを噴射したところシュレッダーが爆発した。	エアゾールスプレー	シュレッダーの屑箱の空間はガスが溜まりやすい場所であるところ、スプレーから潤滑油と共に噴射された可燃性ガスがシュレッダーの内部に残留して、シュレッダーのモーターで発生した火花により引火や爆発を起こす。	潤滑スプレーその他各種スプレーの類をシュレッダーの内部へ向けて噴射してはいけません。	○誤使用→爆発	可燃ガスを検知	緊急通知/サービス依存応答	N:ブザー R1:近づく R2:電源を切る R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/aerosol.pdf
29	浴室の床で市販のガス抜き器を使用してエアゾール缶（LPガスが使用されている）に内容物が残った状態で、缶を開けていたところ、LPガスに何らかの原因で火がつき火傷を負った。	エアゾールスプレー	スプレー缶に内容物が残った状態で、缶に穴を開けてガス抜きをすると、スプレーに用いられている可燃ガス（LPガス）が周囲の火種や金属同士が衝撃的に擦れ合って生じた火花により引火・爆発することがあります。	スプレー缶は中身を使い切ってから捨てましょう。（「中身排出機構」が缶に付いている場合には、それを用いましょう。）缶に釘やガス抜き器などを用いて孔を開けてはいけません。中身を使い切る操作は、風通しが良い火気のない屋外などの場所で行い、噴射音が聞こえなくなるまで繰り返し噴霧用ボタンを押して完全に中身を排出しましょう。	○その他（処分）→発火	可燃ガスを検知	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/aerosol.pdf
30	扉の開閉を繰り返すことにより、徐々に内部の電気配線が切れていき、発熱し出火	ビルトイン式電気食器洗機	経年劣化、内部の電気配線が切れた	点検する	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/choki.pdf
31	天井裏にある電源コードなどに接触不良が生じて発熱し出火	浴室用電気乾燥機	経年劣化、電源コードの接触不良	点検する	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/choki.pdf
32	排気筒のずれ・ほこりの堆積・換気不良による一酸化炭素中毒、	屋内式ガスふろがま（都市ガス/プロパンガス）	経年劣化、排気筒のずれ・ほこりの堆積・換気不良	点検する	経年劣化&保守・設置不良→○CO発生	COを検知する	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/choki.pdf
	排水口のつまりによりふろがまが繰り返し冠水しガス漏れして出火・爆発	屋内式ガスふろがま（都市ガス/プロパンガス）	排水口のつまり	点検する	保守・設置不良&○異物・汚染・吸水→発火	水量レベルを検知	許容値外通知/現場応答	N:ブザー R1:近づく R2:水量が下がる	http://www.meti.go.jp/product_safety/consumer/pdf/choki.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
33	ゴム製の部品が劣化して石油が漏れ、漏れた石油に火がついて出火	石油ふろがま	経年劣化、ゴム製の製品が劣化して石油が漏れた	点検	経年劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
34	ゴム製の部品が劣化して石油が漏れ、漏れた石油に火がついて出火	石油給湯器	経年劣化、ゴム製の製品が劣化して石油が漏れた	点検	経年劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
35	ゴム製の部品が劣化して燃焼ガスが室内に漏れ、不完全燃焼にいたり一酸化炭素中毒	FF 式石油温風暖房機	経年劣化、ゴム製の部品が劣化して燃焼ガスが室内に漏れた	点検	経年劣化→○CO 発生	CO を検知する	緊急通知/サービス依存応答	N:ブザー R1:離れる R2:換気する R3:濃度が下がる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
36	室外機の内部の電気部品(コンデンサー等)が故障して発煙・出火、小動物やほこり・湿気が電気基板部に侵入してトラッキング*現象等が生じて発煙・出火	エアコン	経年劣化	点検	経年劣化→電子部品故障&トラッキング→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
37	電気部品(モーター/コンデンサー/首振り部の電気配線)が劣化してショートし発煙・出火	扇風機	経年劣化	点検	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
38	電気部品(高い電圧を作るトランス等)が劣化して放電し、ほこりや樹脂製部品に着火して出火	ブラウン管テレビ	経年劣化	点検	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
39	電気部品(モーター/コンデンサー)が劣化してショート**し発煙・出火、コンセントの差込プラグ部でトラッキング*現象が生じて出火	換気扇	経年劣化	点検	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
40	電気部品(コンデンサー等)が劣化したり、内部の電気配線が振動により断線して発煙・出火	洗濯機	経年劣化	点検	経年劣化→電子部品故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/cons umer/pdf/c hoki.pdf
41	美容オイルなどが付着した洗濯物を選択し乾燥機を使用、乾燥中の衣類が燃え乾燥機が焼けた。乾燥後のタオルを放置していたら、乾燥機や周辺が焼けた。	乾燥機	洗濯物に残っていたオイルが酸化反応を起こして発熱し、自然発火し乾燥機や周辺を焼いた	美容オイル、食用油、動物油等の付着したものは、乾燥機で乾かさなない。乾燥機にかける場合は油を完全に除去する。洗った後は必ず広げて自然乾燥させる。油が付着したものは重ねておかない。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/oilftyaku.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
42	ガスこんろの下に敷いていた段ボール付近から発火し、台所を焼損した。	ガスコンロ	段ボールがガスバーナーからの放射熱を受けたことにより発火した。炎が直接に可燃物に触れなくともバーナーから可燃物までの距離が近いと発火に至ることがある。	ガスこんろの下に、段ボール、新聞紙、ビニールシートなどの可燃物を敷いてはいけません。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
43	ガスこんろに組み込まれたグリルで魚を焼いていたところ、グリル排気口から炎が上がった。	魚焼きグリル	グリル内部に魚の脂が付着した状態で、使用を続けたためガスバーナーの熱により脂に引火した。	グリル内には、魚の脂などの汚れが付着しないよう、日頃から清掃手入れをする。	保守・設置不良→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
44	揚げ物を調理中、子供に呼ばれたので、ガスレンジ前面にある器具栓を閉じてその場を離れようとした際に、体の一部が鍋に引っかかって揚げ物油の油面が大きく揺らぎ、点火状態のバーナーの近くに油がこぼれてしまったため、油に引火して火災が発生した。(製品には異常はなかった)	ガスレンジ	不注意で、油の入った鍋に体がぶつかり油がこぼれてしまった	揚げ物の調理中は、油の温度などに注意し、そばを離れないようにしましょう。やむをえずガスレンジのそばを離れる際は、あわてずにしっかりと器具栓を閉じてガスの火を消しましょう	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
45	てんぷら油の廃油を処理するためになべに凝固剤を入れたまま加熱し、その場を離れていたところ発火した。	ガスコンロ (てんぷら油の廃油凝固剤)	凝固剤は、冷えた油に入れても溶けないので、油が暖かいうちに(80℃以上)溶かしますが、再加熱すると、油の温度が上がりすぎて発火してしまうことがあります。一般に食用油は加熱を続けると370℃程度で発火しますが、凝固剤を入れると発火温度は30℃～50℃程度下がると言われてい	凝固剤を溶かす際は、調理前に準備しておき、調理が終わった後の油がまだ熱いうちに鍋に入れましょう。もし、冷えた油を再加熱してから溶かす場合は、点火前に凝固剤を入れてから数分の間、様子を見ながらゆっくりかき混ぜ、溶けたら直ぐ火を消します。絶対その場を離れず、加熱しすぎないように注意してください。	○誤使用→発火	加熱中に調理場から離れる	緊急通知/サービス依存応答	N:ブザー R1:近づく	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
46	ガスコンロを使用して天ぶらを調理中にその場を離れていたところ、天ぶら油が発火して火災となった。	ガスコンロ	天ぶら鍋をコンロの火にかけてそのまま放置してしまったため、天ぶら油が過熱されて発火点に達し発火するに至った。	天ぶら調理中は絶対にその場を離れてはいけません。もし離れざるを得ない場合は、必ず消火しましょう。天ぶら油過熱防止装置（あげルック）が装備されているコンロの場合は、天ぶら調理は必ず過熱防止装置がついている側のコンロで調理しましょう。	○誤使用→発火	加熱中に調理場から離れる	緊急通知/サービス依存応答	N:ブザー R1:近づく	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
47	鍋に1cmくらい（約250g）の油を引いて加熱し、過熱状態のまま数分間その場を離れキッチンに戻ると鍋から火柱が上がっていた。	IH調理器	天ぶら調理の際に油の量が少なく、温度調節機能が正常に働かず、油温が上がリすぎる恐れがあります。鍋の底に窪みや反りがあると、温度センサーが正しく検知できなくなる恐れがあります。	油の量は、取扱説明書に指定されている最低限以上の量を入れましょう。天ぶら調理をする場合は、調理器に付属の鍋あるいはメーカー推奨の天ぶら鍋を使用するようにしましょう。調理中はその場を離れないようにしましょう。	○誤使用→発火	加熱中に調理場から離れる	緊急通知/サービス依存応答	N:ブザー R1:近づく	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
48	カレー鍋などを洗っていた食器洗い乾燥機から発煙した。内部を調べたところヒーターカバーに食べ物の残さいとみられる異物が付着していた。	食器洗い乾燥機	ヒーターカバー表面に付着したカレー、油分及び過去から堆積した汚れ分が乾燥動作時のヒーターの加熱により発煙した。	食器や調理器具に残った残さいは、予め手作業で取り除いてから洗浄槽内に入れるようにする。ヒーターカバーには、残さいなどの汚れが付着しないよう日頃から清掃手入れをする。	異物・汚染・吸水&保守・設備不良→○発煙	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
49	プラスチック製の箸箱を食器洗い乾燥機の洗浄かご内に入れてから洗浄を行っていたところ、当該機器から発煙した。	食器洗い乾燥機	洗浄時の水流に飛ばされた箸箱が、食器洗い乾燥機内部のヒーターの上に落ちて加熱されて発煙に至った。	箸箱やプラスチック製容器の蓋など、軽くて小さい食器類は、取扱説明書の指示に従い小物用ポケットに入れて洗うようにしましょう。	誤使用→○発煙	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
50	電気ポットの残り湯の排水時に上蓋を外さずに本体と蓋を持って残り湯を排水したときに、蓋が外れてお湯が太ももに掛かり火傷した。	電気ポット	上蓋が付いたまま湯を捨てると上蓋が外れることがあります。	残り湯を捨てる際は、必ず上蓋を外し、取扱説明書に指示されている方向にポット本体を持って残り湯を捨てるようにしましょう。	誤使用→やけど	×			http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
51	豆が入ったおかゆを調理し、炊飯完了のお知らせ音を聞いてから蓋を開けたところ内容物（おかゆ）が飛び散って火傷した。	圧力炊飯器	豆の皮が調圧部（調圧弁、安全弁）の穴を塞いでしまったために、内部の圧力が高くなってしまい、その状態の時に蓋を開けたため内容物が飛び散った。	圧力炊飯器では豆料理をしてはいけません。	○誤使用→やけど	高い圧力を検知	緊急通知/サービス依存応答	N:ブザー R1:十分時間を置くまで近づかない	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
52	こんろのそばを通った時に体や荷物などがつまみに当たって意図せずスイッチが入り、こんろの上にあった可燃物に延焼。	小形キッチンユニット用電気こんろ	電気こんろの上や周辺に可燃物を置いている。電気こんろの上や周辺には絶対に可燃物を置かない。つまみ部分にカバーがなく露出しているため、意図せずスイッチが入ってしまう。	電気こんろの上や周辺には絶対に可燃物を置かない。つまみ部分にカバーがなく露出しているため、意図せずスイッチが入ってしまう。電気こんろを使用している方は、メーカー等が行う無償改修を受けてください。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/daido_koro.pdf
53	電子レンジの庫内から異音がして出火した。	電子レンジ	庫内に食品カスが付着していました。食品カスに電波が集中して炭化し、発火したためにカバーが焼けたものです。	庫内やドアに汚れが付着すると発火するので、清掃してください。食品の過熱は発煙・発火の原因になるので、注意してください。	異物・汚染・吸水&保守・設備不良→○発煙	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kitchen.pdf
54	ハンドミキサーの刃の内側に詰まった調理物を取ろうとしたところ、指を切った。	ハンドミキサー	電源プラグを抜かず、電源の入った状態で料理用ハンドミキサーの刃の内側につまった材料を取ろうとしたためです。	電源の入った状態で手やスプーン、ヘラなどで触れないでください。使っていないときは、電源プラグを抜いてください。	誤使用→けが	×			http://www.meti.go.jp/product_safety/consumer/pdf/kitchen.pdf
55	揚げ物調理中、その場を離れていたら、温度センサーが作動せずに、なべから発火した。	IH 調理器	なべ底に反りやくぼみがあったため、温度センサーが温度を正常に検知することができなくて発火したものです。	なべ底に反りやくぼみがあると温度センサーが正常に温度を検知できません。底が平らなものを使ってください。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kitchen.pdf
56	ガスこんろのアルミ製汁受けを使用して発火	IH 調理器	点火不良や異常燃焼により火災になることがあります。	点火部や調理油過熱防止装置などをふさがないように適切に装着してください。	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kitchen.pdf
57	I H 調理器の汚れ防止マット使用で、マットや天ぶらなべの油が異常過熱で発火	IH 調理器	温度センサーが温度を正常に検知できなくなり発火につながる場合があります。	調理中はマットを近づけない	誤使用→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/kitchen.pdf
58	スマートフォンの A C アダプターとマルチタップの接続部が焦げた。	スマートフォン	電源プラグとマルチタップの隙間に金属製の異物が入り込んだためにショートしたものです。	コネクタに金属や、汗、飲料水、ペットの尿など液体を付着させない。	誤使用&異物・汚染・吸水→電子部品・故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/sunaho.pdf
59	携帯電話機と A C アダプターとの接続部が溶けた。	スマートフォン	携帯電話機の接続端子内部に塩素を含む液体が付着したためにショートしたものです。	コネクタに金属や、汗、飲料水、ペットの尿など液体を付着させない。	誤使用&異物・汚染・吸水→電子部品・故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/sunaho.pdf
60	スマートフォンの A C アダプターから火花が出て、指にやけどを負った。	スマートフォン	コネクタに無理な力を加えて抜き差ししていたために変形が生じ、ショートしたものです。	コネクタを接続するときは無理な力をいれない。一度変形したコネクタは使用しない。	誤使用&異物・汚染・吸水→電子部品・故障→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/sunaho.pdf
61	ペットによる事故も発生しています。スマートフォンとペットに関わる火災	スマートフォン	尿をかけたり、本体をかんだために内蔵している電池が破裂し、火災に	ペットの届かないところにスマートフォンを管理する	○保守・設置不良→電子部品故障→発火	ペット と スマートフォンが近づく	呼び出し通知/現場応答	N: ブザー R1: 近づく R2: ペットとスマートフォンの距離を取る	http://www.meti.go.jp/product_safety/consumer/pdf/sunaho.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
62	便座や便座コードから火が出る場合があります。	温水洗浄便座	便座のゴム足が外れている、ガタツキがある。便座コードがねじれたり、便座で挟み込んだりしている。便座にひびや割れがある。便座が異常に熱いときや、冷たいときがある。	すぐに電源プラグを抜いて、止水栓を閉めてください!点検を依頼してください!	保守・設置不良→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
63	製品内部の電子・電気部品に水がかかり、製品から火が出る場合があります。	温水洗浄便座	製品から水漏れしている。操作部のシールがめくれたり、ひび割れたりしている。	すぐに電源プラグを抜いて、止水栓を閉めてください!点検を依頼してください!	経時劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
64	電源コードやコンセント部から火が出る場合があります。	温水洗浄便座	電源コードが熱くなっている。電源プラグの差込部が発熱・変色している。	すぐに電源プラグを抜いて、止水栓を閉めてください!点検を依頼してください!	経時劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
65	熱いお湯が出てやけど	温水洗浄便座	水漏れ	点検	経時劣化→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
66	漏電による感電	温水洗浄便座	水漏れ	点検	経時劣化→○感電	漏電を検知	緊急通知/サービス依存応答	N:ブザー R1:近づくと R2:電源を切る	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
67	階下への漏水	温水洗浄便座	水漏れの継続	点検	経時劣化→○漏水	漏水を検知	緊急通知/サービス依存応答	N:ブザー R1:近づくと R2:水を片付ける	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
68	感電、火災	温水洗浄便座	間違った洗剤の使い方（プラスチック部分は専用クリーナーや薄めた中性洗剤を柔らかい布に含ませて拭き、さらに水拭きをしてください。便器用洗剤は、3分以内に洗い流す）	プラスチック部分は専用クリーナーや薄めた中性洗剤を柔らかい布に含ませて拭き、さらに水拭きをしてください。便器用洗剤は、3分以内に洗い流す	誤使用→○感電、発火	発煙、漏電を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
69	低温やけど	温水洗浄便座	便座に長時間皮膚が触れている、温風乾燥を同じ場所に長く当てている	温度調節などにご注意ください。	○誤使用→やけど	長時間使用を検知	許容値外通知/現場応答	N:ブザー R1:便座から離れる R2:温度調整する	http://www.meti.go.jp/product_safety/consumer/pdf/onsuisenjoybenza.pdf
70	出火事故	ウォータースーバ	電源コード、電源プラグ、電気部品の異常、ホコリがたまっている	コンセントを抜き、容器を外し、点検を受ける	異物・汚染・吸水&保守・設備不良→○発火	発煙を検知	緊急通知/サービス依存応答	N:ブザー R1:離れる	http://www.meti.go.jp/product_safety/consumer/pdf/water.pdf
71	やけど	ウォータースーバ	小さい子供が温水コックに触れる	チャイルドロック付きのものを使う。ウォータースーバをゆすったり倒したりしない。ウォータースーバの移動は、コンセントを抜いてから30分以上放置してから。	誤使用→やけど	×			http://www.meti.go.jp/product_safety/consumer/pdf/water.pdf

table continued on next page

id	事例名	機器	原因	対策	事故経路と通知のタイミング	トリガー	通知と応答の種類	通知と応答例	url
72	ハリケーン時に発電機を屋内で使用し一酸化炭素中毒になる	Portable Generator(ガソリン発電機)	酸素量の不足により不完全燃焼が起こったため	発電機は外で使う、ドアから 20 フィート離して使う、雨が降りやむのを待ってから使う	○誤使用→CO 発生	屋内での使用を検知	許容値外通知/復帰応答	N:ブザー R1: 近づく R2: 電源を切る R3: 外に動かす	https://www.cpsc.gov/s3fs-public/Hurricane%20tri-fold%20082818_0.pdf?eTnpY87MgYzm0sSUkoT9XIM1q3uV41Cz
73	濡れた環境で発電機を使い感電する	Portable Generator(ガソリン発電機)	水分を伝って電気が流れた	湿気の多い場所で使わない、水たまりのある場所で使わない、ぬれた手で触らない	○誤使用→感電	湿気を感じ	許容値外通知/現場応答	N:ブザー R1: 近づく R2: 電源を切る R3: 湿気が収まる	https://www.cpsc.gov/s3fs-public/Portable_Generator_Safety_Alert_2017_5123.pdf?5zYC5oSH8WSNeBNsFMGm1L0J0.s1W07a
74	発電機の電力を家の配線につなげたところ、事業者や近隣住民が感電したり、家庭用回路保護装置を短絡させた	Portable Generator(ガソリン発電機)	backfeeding(逆給電)が起こった。電力線に再通電され、触っている人が感電した。	直接接続しない。専用の住宅用発電機を使う。	○誤使用→感電&発火	逆給電を感じ	許容値外通知/現場応答	N:ブザー R1: 近づく R2: 電源を切る	https://www.cpsc.gov/s3fs-public/Portable_Generator_Safety_Alert_2017_5123.pdf?5zYC5oSH8WSNeBNsFMGm1L0J0.s1W07a

付 録 B 実装例

実装例は、「炊飯器の炊き上がり時の動作完了通知」である。通知と応答の組み合わせは炊飯器の取扱説明書¹から図 B.1 のように判断する。通知そのものの発生は取扱説明書の「ピーピーピーピー」というブザー音の記載から判断する。通知の発生条件は「炊き上がり」という動作の終了を表す記載から判断する。通知後のユーザの行動、つまりユーザの応答は「ごはんをよくほぐす」という記載から判断する。

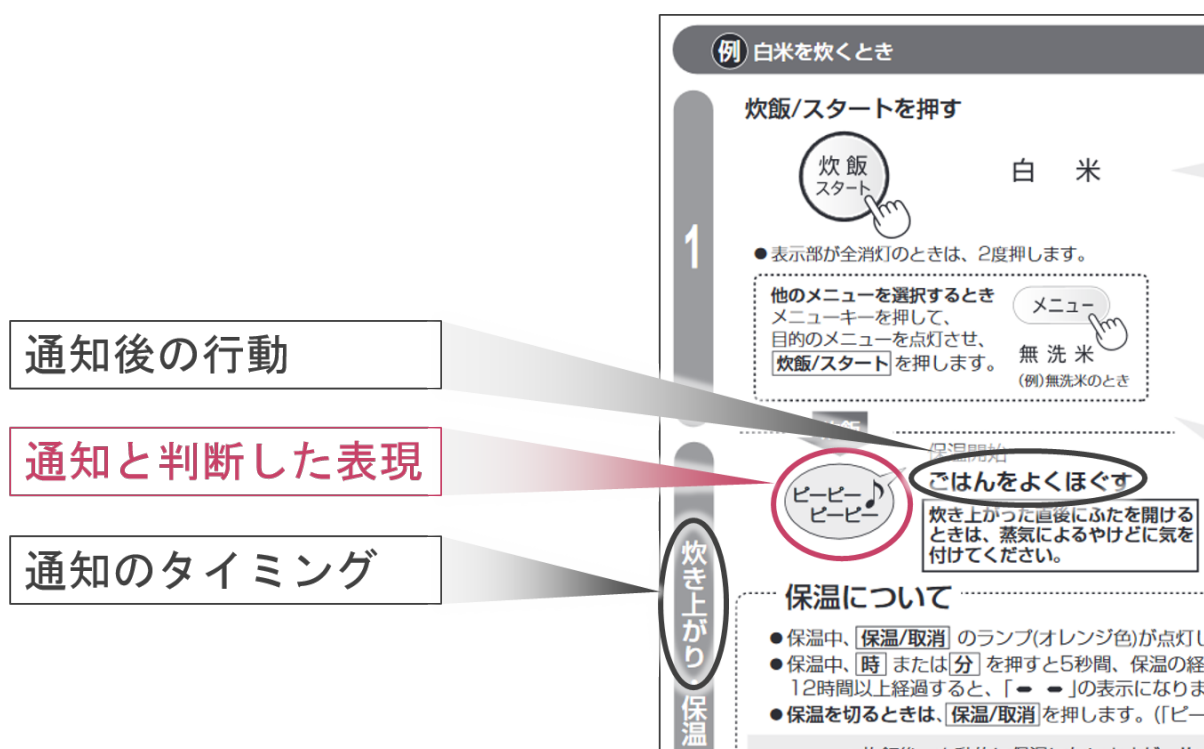


図 B.1: 炊飯器の取扱説明書

図 B.2 は、2.3 節で示した通知と応答のマトリックス (図 2.6) 上に、「炊飯器の炊き上がり時の動作完了通知」を要求するアプリケーションが API を通して実行すべき引数を重ね合わせたものである。この場合は、通知種が動作終了通知であり、応答種は暗黙応答及び現場応答となる。

¹http://www.sharp.co.jp/support/ricecooker/doc/kscf05a_mn.pdf?productId=KS-CF05A

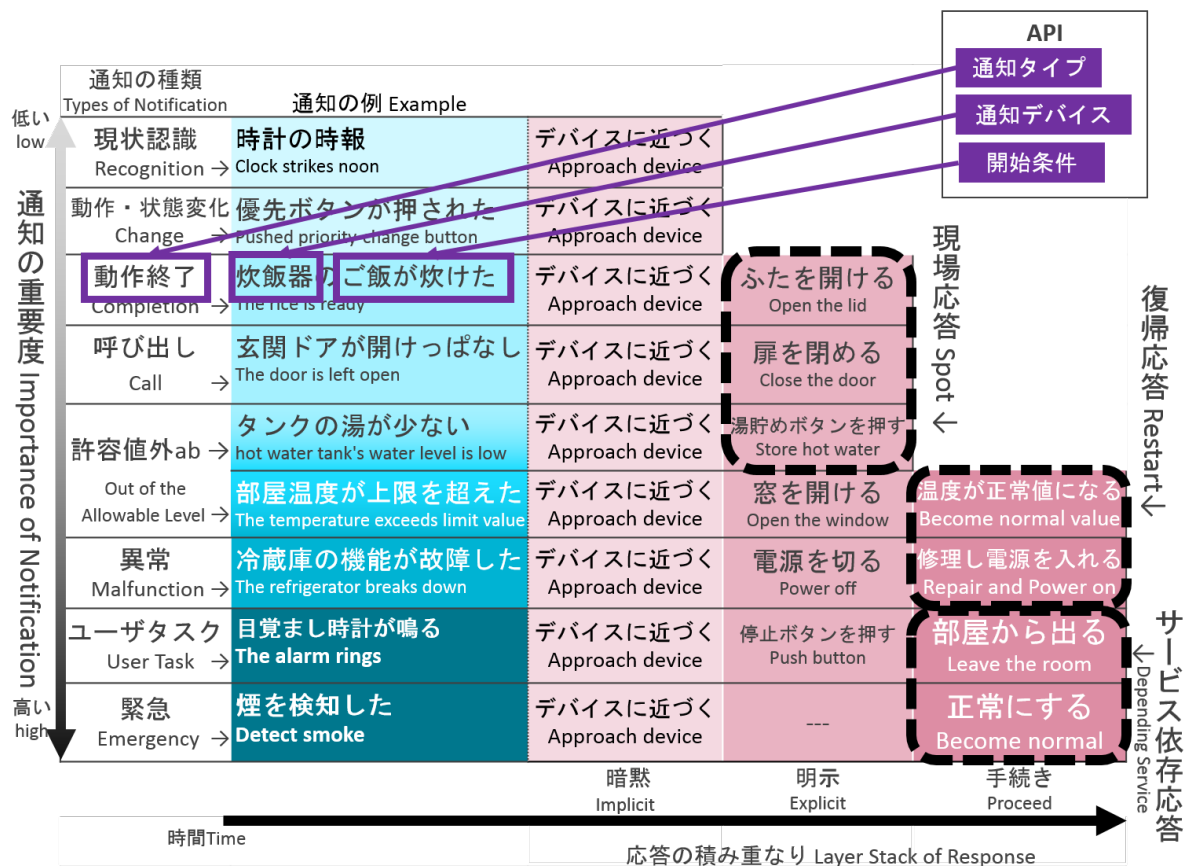


図 B.2: 通知と応答のマトリックスと、炊飯器の対応関係

今回の実装例で双方向通知応答基盤が行うべき役割は図 B.3 となる。アプリケーションは基盤に通知を開始してもらい、その結果として応答の成否をもらう。また、ユーザは基盤に繋がった炊飯器のブザーからアナウンス音を聞き、炊飯器に近づき、炊飯器のふたを開ける。

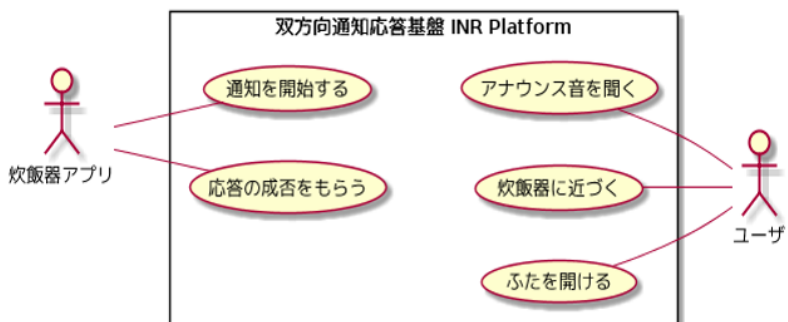


図 B.3: 炊飯器の炊き上がり時の動作完了通知のユースケース図

実機の構成は次の図 B.4 となる。用いた器具は次のとおりである。

- RaspberryPi2 TypeB
- ダンボール製の炊飯器を模したモデル
- 傾き (開閉) センサ SW520D
- 接近センサ HC-SR501
- ブザー HDB06LFPN

メインPCとデバイス管理をする RaspberryPi2 は Ethernet による接続をしている。RaspberryPi2 と炊飯器モデルのセンサ及びブザーは GPIO による接続をしている。RaspberryPi2 では EchonetLite デバイスを構築できる humming²を用いている。これによってセンサの生データを RaspberryPi2 上で EchonetLite の仕様に沿った形式に変更でき、RaspberryPi のコンソールログ上で EchonetLite デバイスとして発見したことを確認できる。

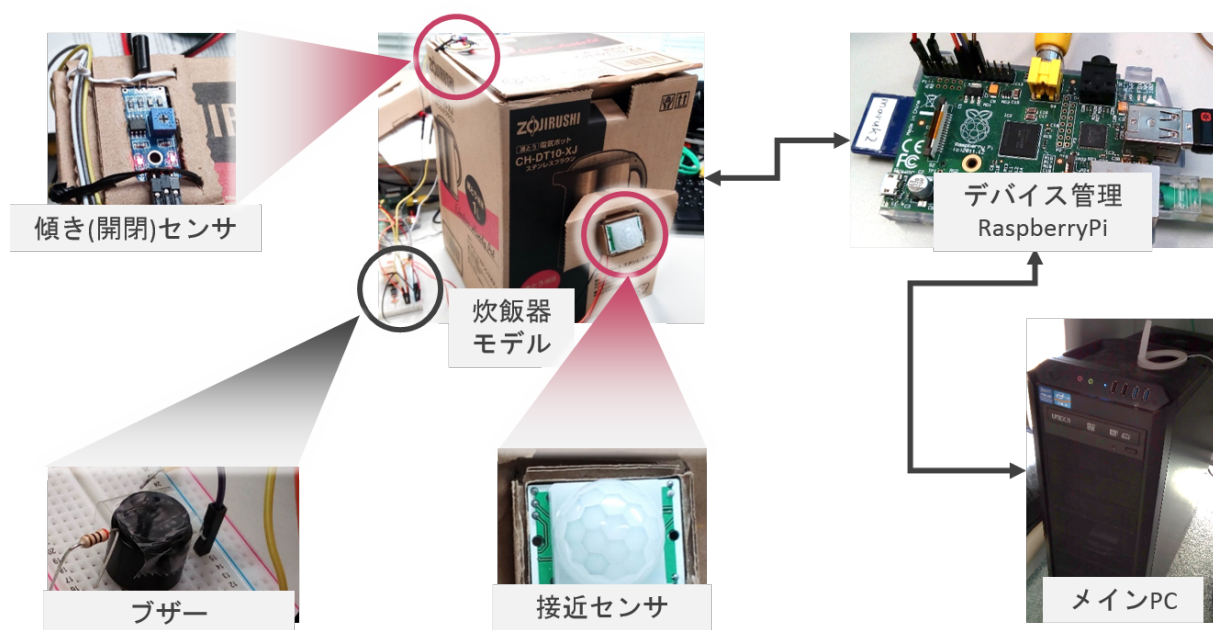


図 B.4: 実機の構成

²<https://github.com/ymakino/humming>

今回の実装例におけるシーケンスを図 B.5 に示す。まず、炊飯器アプリから基盤に通知要求を出す。基盤は炊飯器に取り付けた位置センサに対してユーザの接近を感知する命令を出しておき、その後に、炊飯器内臓のブザーに音を鳴らす命令を出す。炊飯器内蔵のブザーはユーザに音を伝える。位置センサがユーザの接近を感知する。位置センサは基盤に接近があったことを知らせ、基盤内部で暗黙応答が完了したという情報を更新する。ユーザが炊飯器のふたを開けたことを炊飯器内臓の開閉センサが感知する。開閉センサは基盤に開閉があったことを知らせ、基盤内部で現場応答が完了したという情報を更新する。そして、基盤は炊飯器アプリに応答が完了したことを伝える。

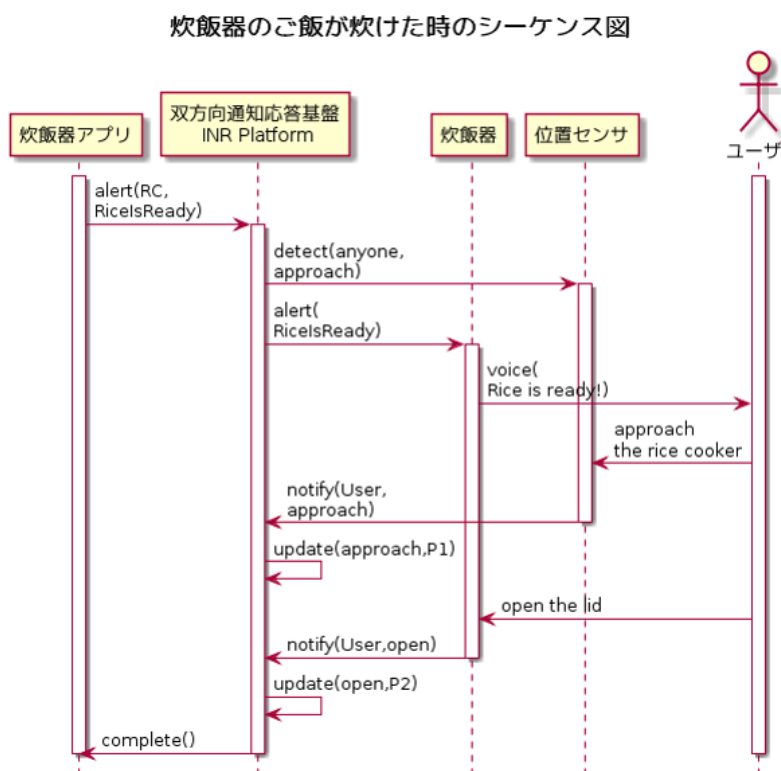


図 B.5: 炊飯器の炊き上がり時の動作完了通知のシーケンス図

本研究では通知応答を行うデバイス及びユーザのリアルタイムな動作及び行動を時間制約付き状態遷移モデルを用いて実装を行った。これは、シーケンス図 B.5 で表した双方向通知応答基盤のライフラインが実行状態である時、他のオブジェクトに対する命令をどのような順序で行うか決定するためである。UPPAAL³は状態遷移を GUI でモデリングでき、時間制約を満たしているかモデル検査できる⁴ソフトウェアである。今回モデル検査は行わないが、制約違反するパスが存在しないような状態遷移モデルの作成が可能になる。図 B.6 は UPPAAL で用いるモデルの凡例である。Trigger は、遷移に関連するイベントを表す。Guard は、記述した条件が満たされる場合のみ遷移するパスを表す。Effect は、実行するアクション (関数、代入処理など) を表す。Select は、遷移時に数値を範囲から選択し変数に代入する。Committed Location は、時間経過や他の遷移よりも優先されるノードを表す。Trigger? 及び Trigger! について、? から ! へイベント (フラグ) を伝達することを表す。

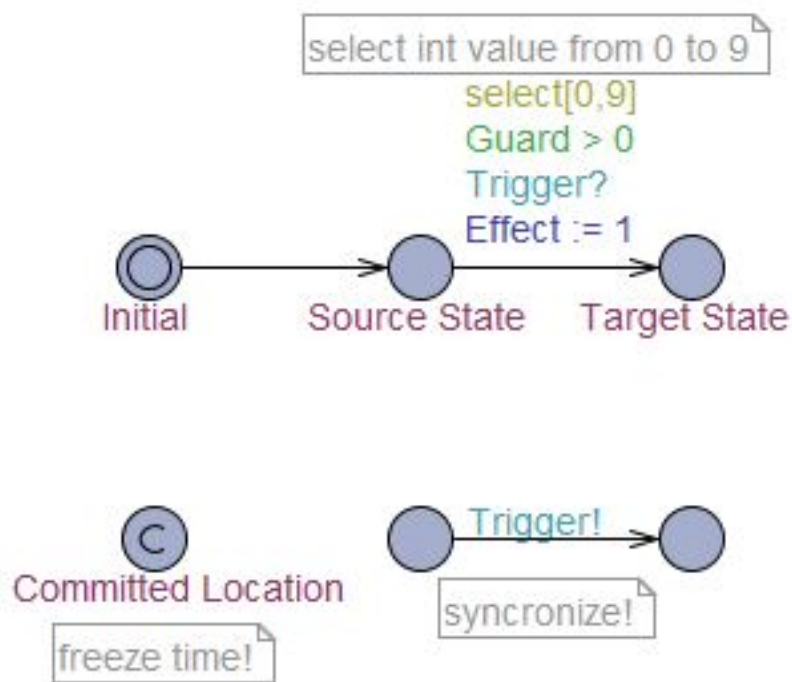


図 B.6: UPPAAL で用いる状態遷移モデルの凡例

³<http://www.uppaal.org/>

⁴長谷川哲夫, 田原康之, and 磯部祥尚. "UPPAAL による性能モデル検証—リアルタイムシステムのモデル化とその検証 (トップエスイー実践講座)." (2012).

図 B.7 は今回実装する「炊飯器の炊き上がり時の動作完了通知」の状態遷移図である。中央の Platform は「炊飯器の炊き上がり時の動作完了通知」時の状態遷移であり、他の全ての状態遷移を中継する機能がある。左上の Buzzer は炊飯器に設置したブザーの状態遷移を表す。中央上の HumanDetectionSensor は炊飯器に設置した接近センサの状態遷移を表す。右上の OpenCloseSensor は炊飯器に設置したふた開閉センサの状態遷移を表す。左下の Smartphone は炊飯器アプリ及び位置測位機能があるスマートフォンの状態遷移を表す。なお今回の実装では、位置測位機能は常に屋内に固定してある。右下の User は「炊飯器の炊き上がり時の動作完了通知」に関わるユーザの状態遷移を表す。Platform・Smartphone・User は屋内と屋外両方を想定した状態遷移となっているが、プログラム上では屋内の場合のみを実装した。

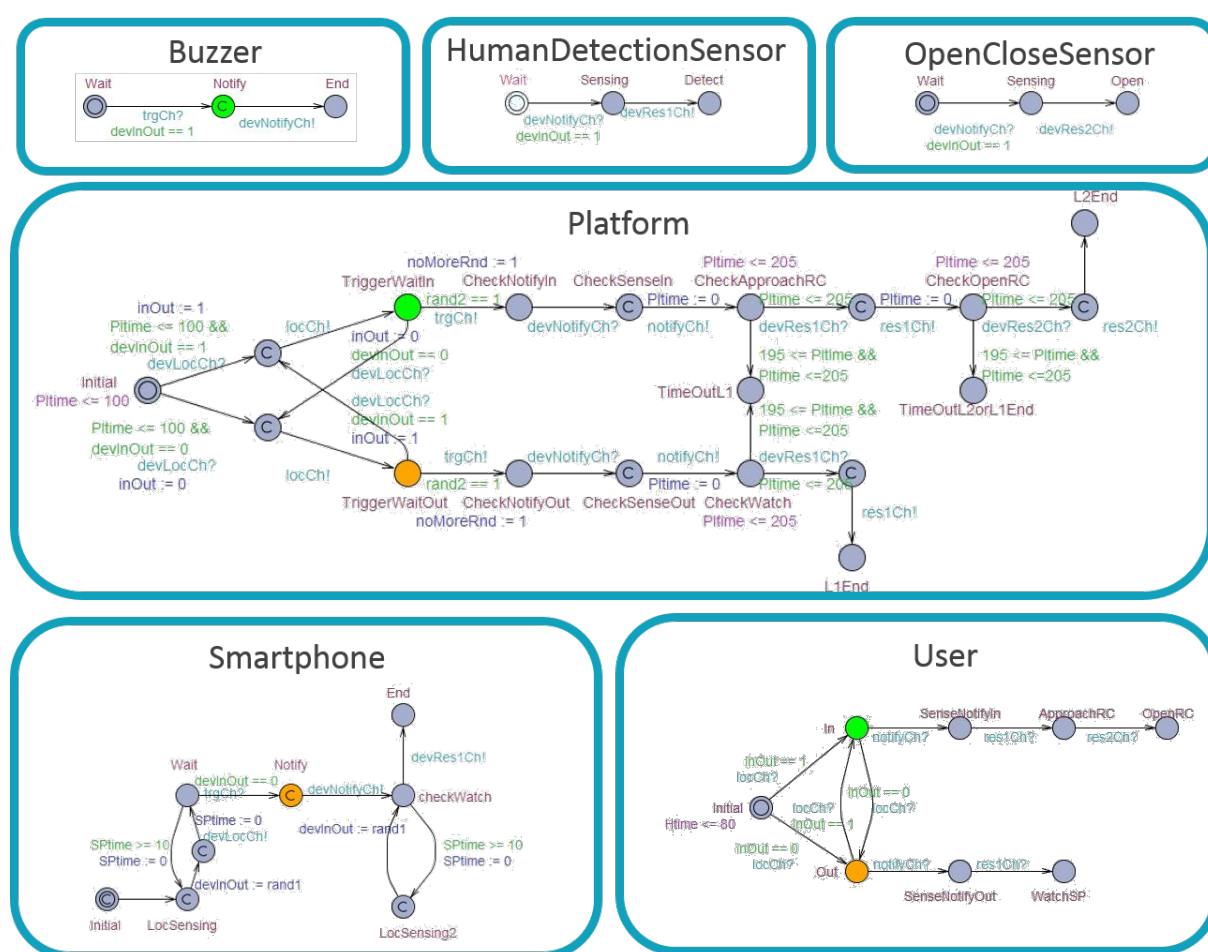


図 B.7: 「炊飯器の炊き上がり時の動作完了通知」の状態遷移

状態遷移の実際の動作について、図 B.7 を分割しながら説明する。

図 B.8 は Platform の左半分と Smartphone の連携を表す。Smartphone は LocSensing ノードから Wait ノードへ移る時、devLocCh!トリガを Platform に伝播し状態の遷移を促す。Platform は devLocCh? でトリガを受け付けると、devInOut の値に応じて Initial ノードから TriggerWaitIn または TriggerWaitOut に遷移する。なお、devInOut が 0 の時は測位データが屋外であることを表し、1 の時は屋内を表す。

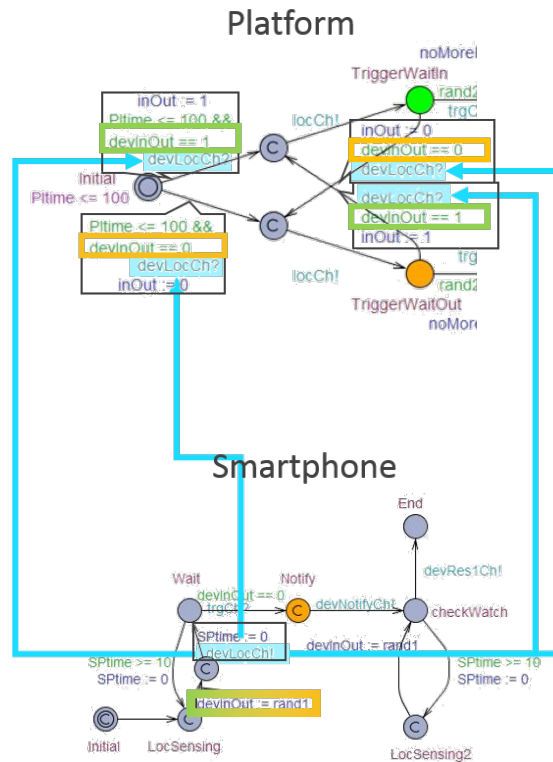


図 B.8: デバイスの位置情報の伝達

図 B.9 は Platform の左半分と User の連携を表す。Platform は Initial ノードから TriggerWaitIn または TriggerWaitOut ノードへ移る時、locCh!トリガを User に伝播し状態の遷移を促す。User は locCh?でトリガを受け付けると、inOut の値に応じて Initial ノードから In または Out ノードに遷移する。なお、inOut が 0 の時はユーザが屋外に居ることを表し、1 の時は屋内に居ることを表す。

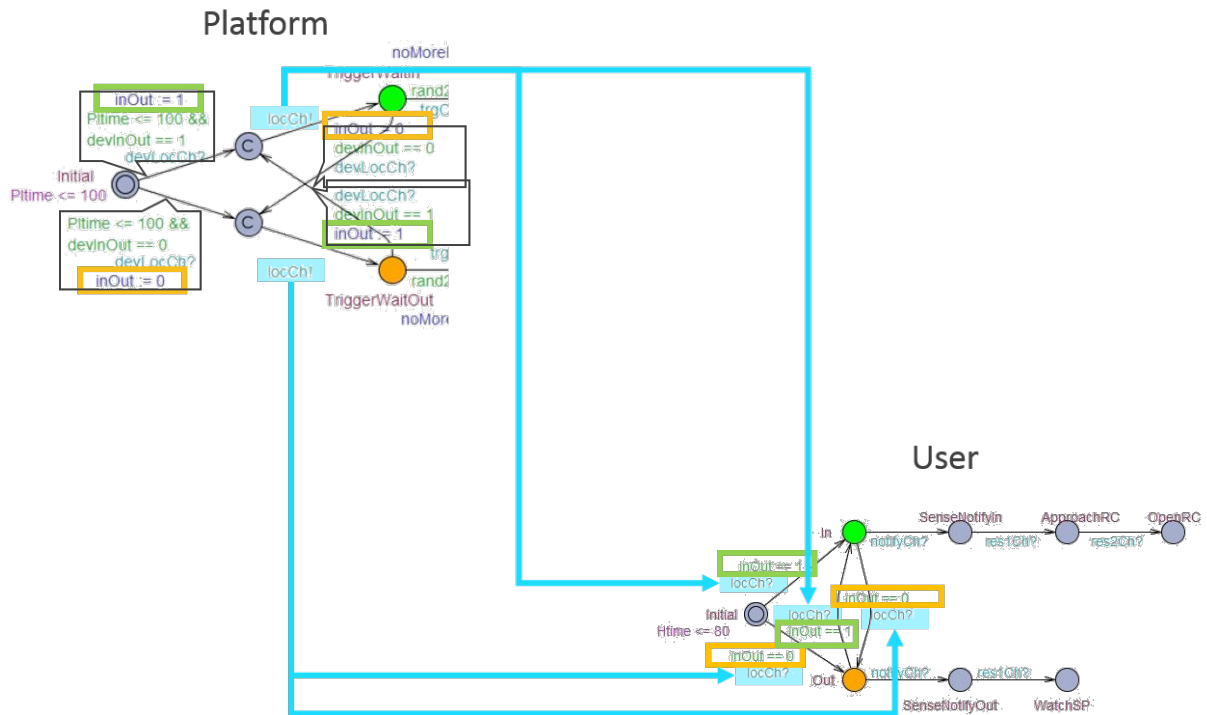


図 B.9: ユーザの位置情報の伝達

図 B.10 は Platform・Buzzer・Smartphone の連携を表す。Platform が通知の要求を受け付けた時、trgCh! を Buzzer または Smartphone に伝播し状態の遷移を促す。屋内に居る時、Buzzer は trgCh? でトリガを受け付けると、Wait ノードから Notify ノードに遷移する。また屋外に居る時、Smartphone は trgCh? でトリガを受け付けると、Wait ノードから Notify ノードに遷移する。

Notify ノードに遷移した時、StateMachine 系から Resource 系に、そして Resource 系から Device 系に通信が発生する。Device 系は eBuzzer クラスまたは eSmartphone クラスを通じて音声を発生させる。

なお、本来は炊飯器アプリとしての機能を持つはずの Smartphone は、Platform に対して通知の要求を出すトリガを持ち、その情報を伝播すべきである。

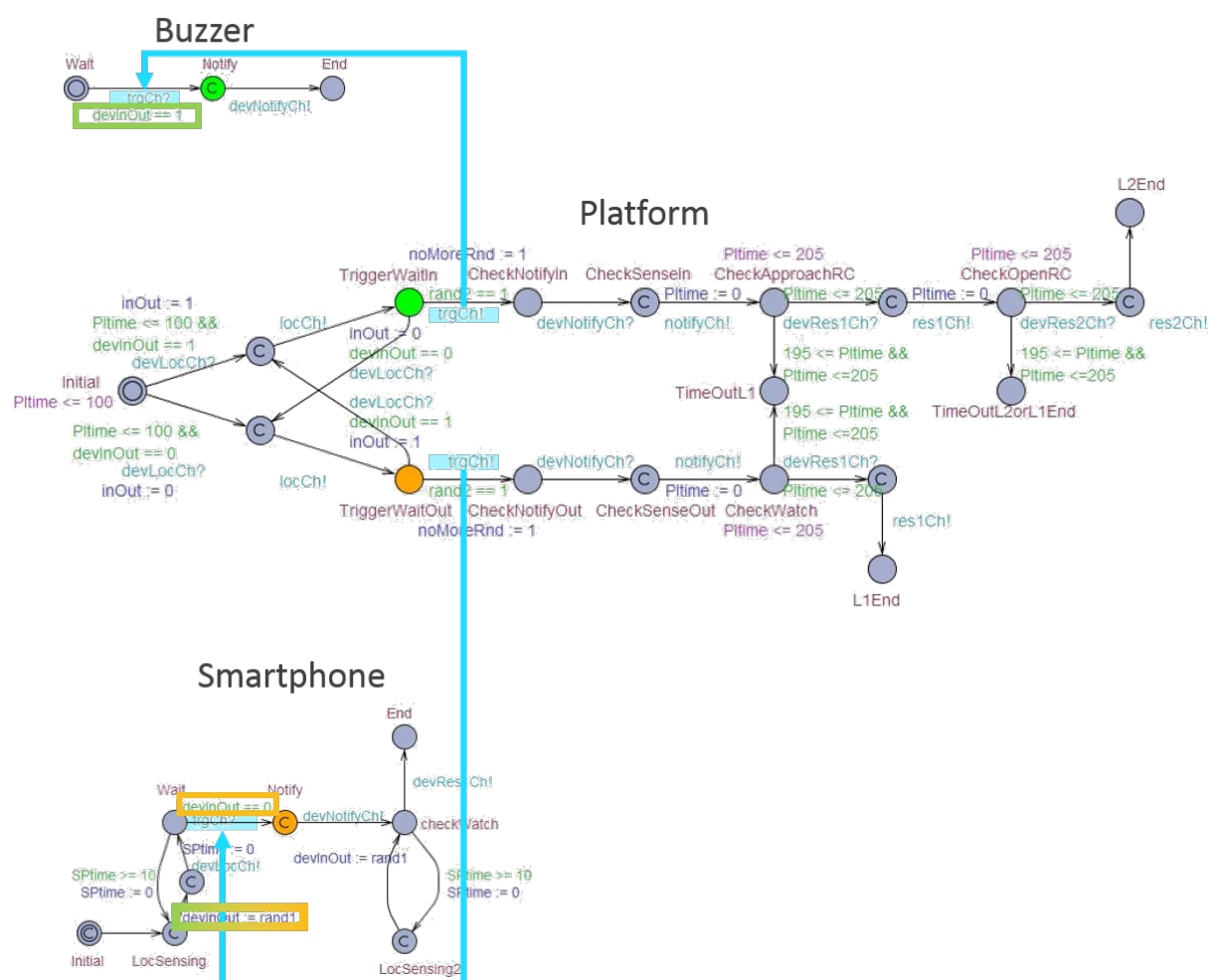


図 B.10: 通知要求の伝達

図 B.11 は Platform・Buzzer・HumanDetectionSensor・OpenCloseSensor・Smartphone の連携を表す。屋内に居る時、Buzzer が Notify ノードから End ノードに遷移する時、devNotifyCh!トリガを Platform・HumanDetectionSensor・OpenCloseSensor に伝播し状態の遷移を促す。HumanDetectionSensor は devNotifyCh?でトリガを受け付けると Wait ノードから Sensing ノードに遷移する。OpenCloseSensor も同様に devNotifyCh?でトリガを受け付けると Wait ノードから Sensing ノードに遷移する。Platform は devNotifyCh?でトリガを受け付けると ChekcNotifyIn ノードから CheckSenseIn ノードへ遷移する。

また屋外に居る時、Smartphone が Notify ノードから checkWatch ノードに遷移する時、devNotifyCh!を Platform に伝播し状態の遷移を促す。Platform は devNotifyCh?でトリガを受け付けると ChekcNotifyOut ノードから CheckSenseOut ノードへ遷移する。

HumanDetectionSensor が Sensing ノードに遷移した時、StateMachine 系→Resource 系→Device 系に通信が発生し、接近センサがセンシングを始める。同様に OpenCloseSensor も Sensing ノードに遷移した時、開閉センサがセンシングを始める。Smartphone も checkWatch ノードに遷移した時、通知画面のタップのセンシングを始める。

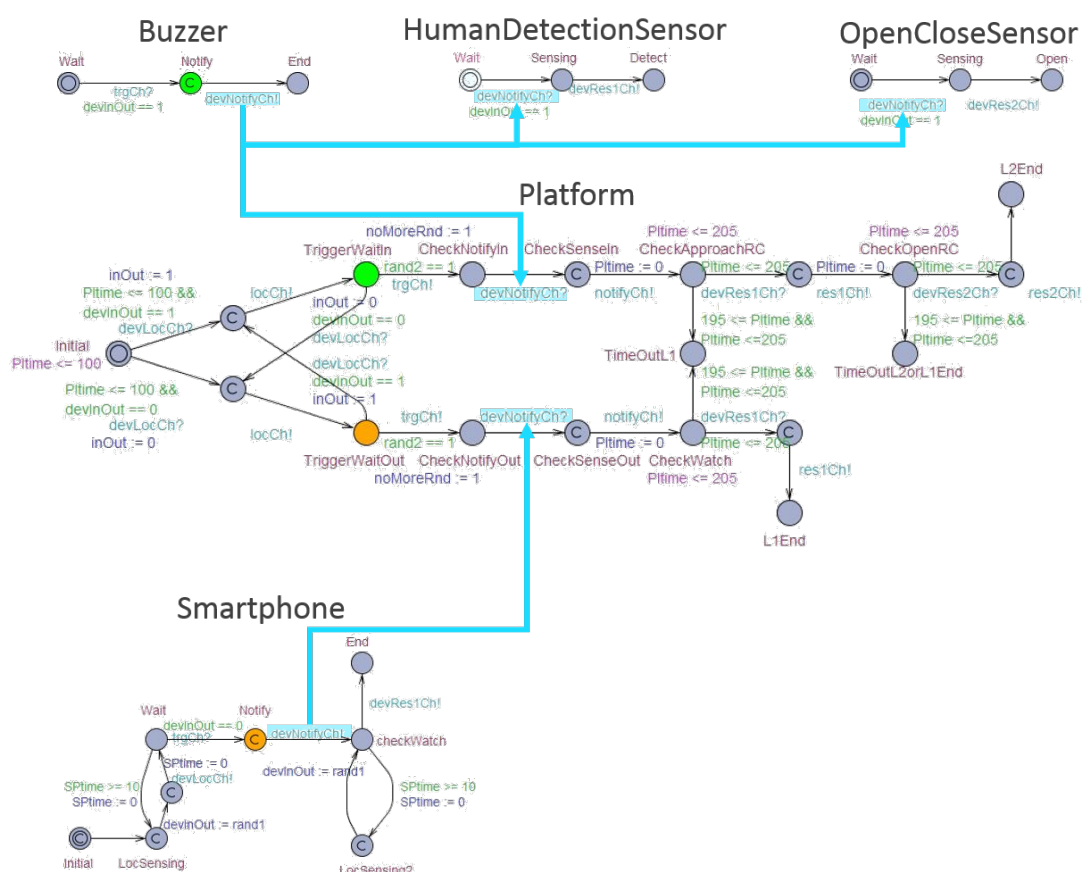


図 B.11: 通知後の応答開始要求の伝達

図 B.12 は Platform と User の連携を表す。Platform が CheckSenseIn または CheckSenseOut ノードから CheckApproachRC または CheckWatch ノードに遷移する時、notifyCh! トリガを User に伝播し状態の遷移を促す。User は notifyCh? でトリガを受け付けると、In または Out ノードから SenseNotifyIn または SenseNotifyOut ノードに遷移する。

SenseNotifyIn または SenseNotifyOut は反射的な応答を表すが、今回はその反応を必ず行ったと仮定する。もし耳を澄ます動作や、体を傾けるといった反射的な行動をセンシングできるならば、そうしたセンサの状態遷移と連携すべきである。

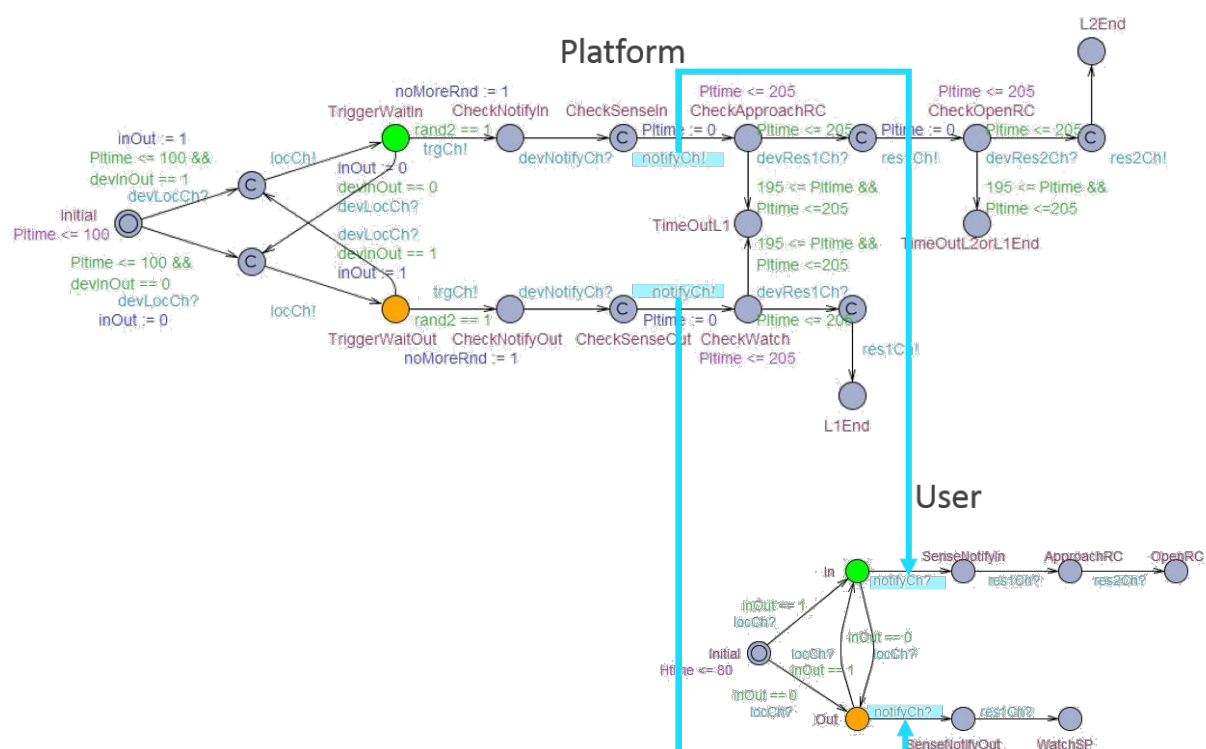


図 B.12: Platform から User への通知を受け取ったことの伝達

図 B.13 は Platform・HumanDetectionSensor・Smartphone の連携を表す。屋内に居る時、HumanDetectionSensor が Sensing ノードから Detect ノードに遷移する時、devRes1Ch!トリガを Platform に伝播し状態の遷移を促す。Platform は devRes1Ch? でトリガを受け付けると、CheckApproachRC ノードから CheckOpenRC ノードに遷移する。

屋外に居る時、Smartphone が checkWatch ノードから End ノードに遷移する時、devRes1Ch!トリガを Platform に伝播し状態の遷移を促す。Platform は devRes1Ch? でトリガを受け付けると、CheckWatch ノードから L1End ノードに遷移する。

Device 系で接近を感知した時、Device 系→Resource 系→StateMachine 系で通信が発生し、HumanDetectionSensor の状態は Sensing ノードから Detect ノードへと遷移する。同様に Smartphone の状態は checkWatch ノードから End ノードへと遷移する。

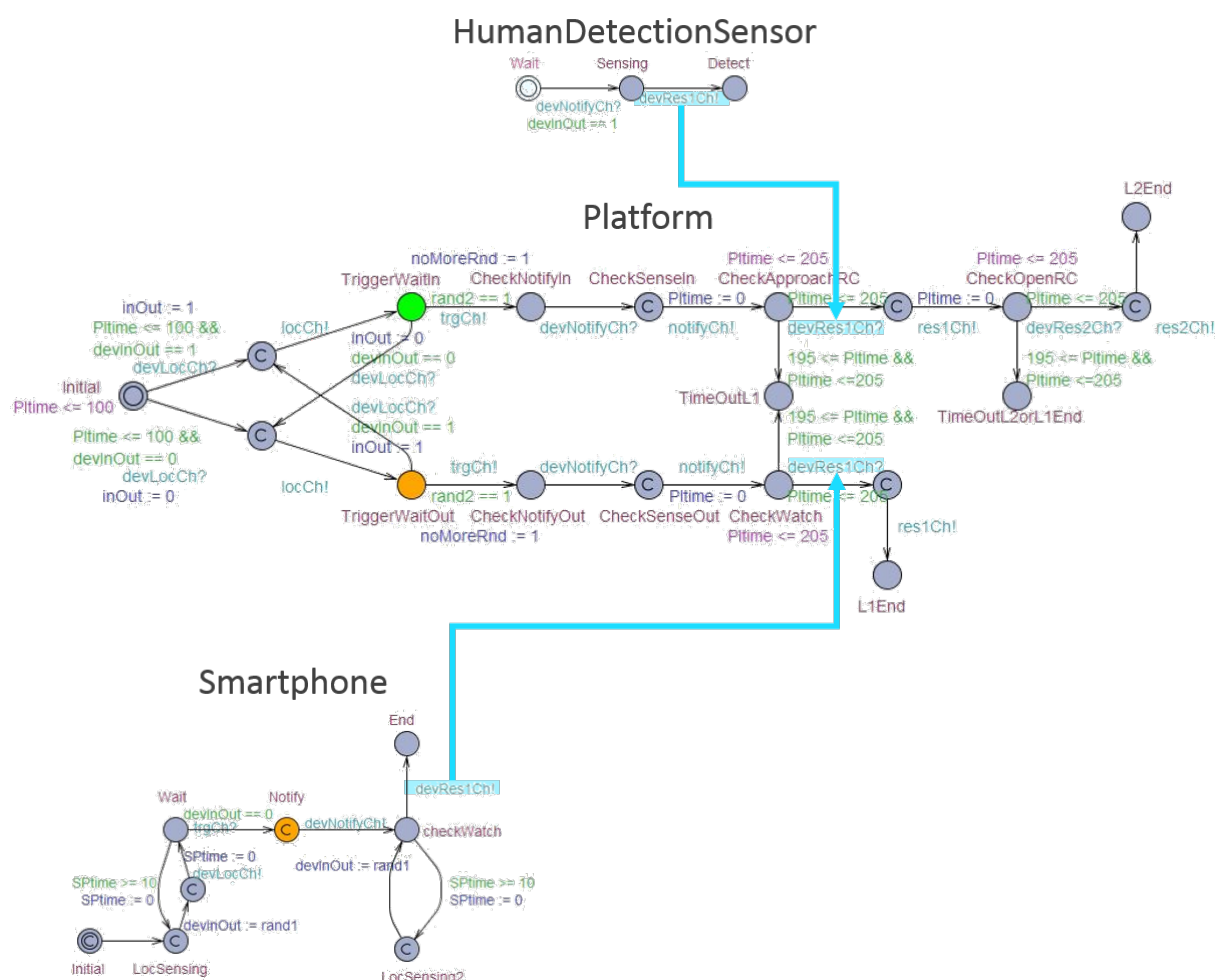


図 B.13: 暗黙応答をセンサが感知した時の伝達

図 B.14 は Platform と User の連携を表す。Platform が CheckApproachRC または CheckWatch ノードから CheckOpenRC または L1End ノードに遷移する時、res1Ch!トリガを User に伝播し状態の遷移を促す。User は res1Ch?でトリガを受け付けると、SenseNotifyIn または SenseNotifyOut ノードから ApproachRC または WatchSP ノードに遷移する。

User の ApproachRC ノードは暗黙応答としてユーザが炊飯器に近づいたことを表す。また、WatchSP ノードは暗黙応答としてユーザが画面を思わずタッチしたことを表す。

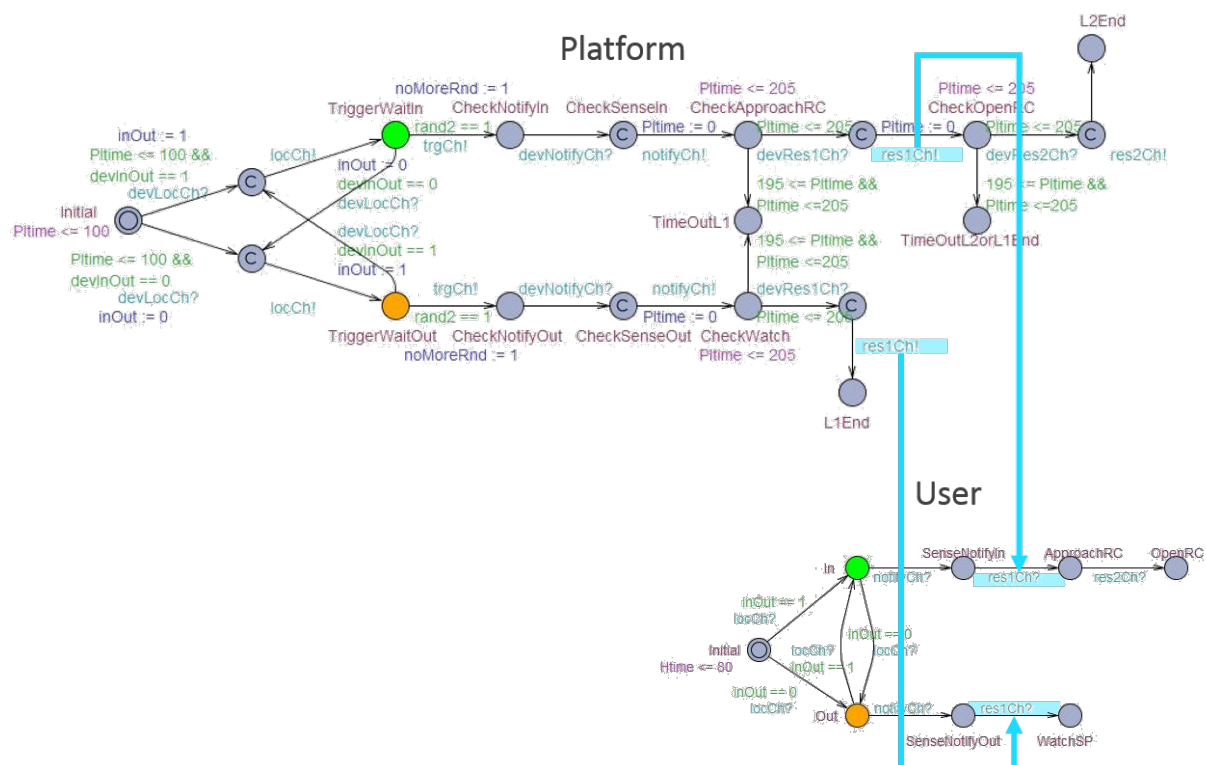


図 B.14: 暗黙応答完了の伝達

図 B.15 は Platform と OpenCloseSensor の連携を表す。OpenCloseSensor が Sensing ノードから Open ノードに遷移する時、devRes2Ch!トリガを Platform に伝播し状態の遷移を促す。Platform は devRes2Ch? でトリガを受け付けると、CheckOpenRC ノードから L2End ノードに遷移する。

Device 系でふたの開閉した時、Device 系→Resource 系→StateMachine 系で通信が発生し、OpenCloseSensor の状態は Sensing ノードから Open ノードへと遷移する。

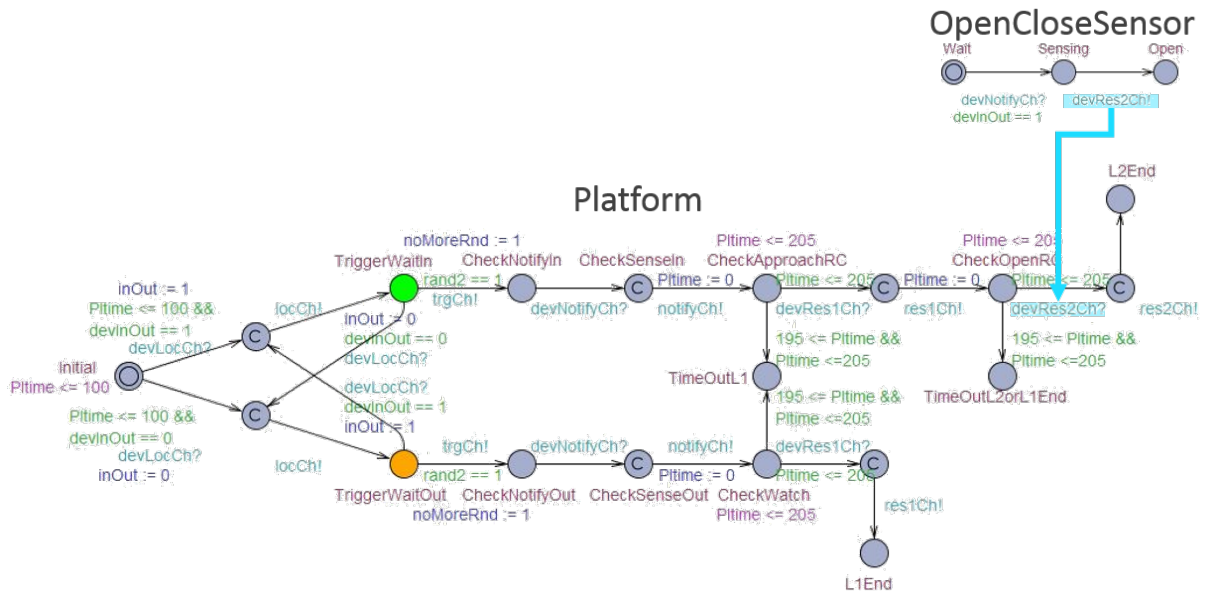


図 B.15: 現場応答をセンサが感知した時の伝達

図 B.16 は Platform と User の連携を表す。Platform が CheckOpenRC ノードから L2End ノードに遷移する時、res2Ch!トリガを User に伝播し状態の遷移を促す。User は res2Ch? でトリガを受け付けると、ApproachRC ノードから OpenRC ノードに遷移する。

User の OpenRC ノードは現場応答としてユーザが炊飯器のふたを開けたことを表す。

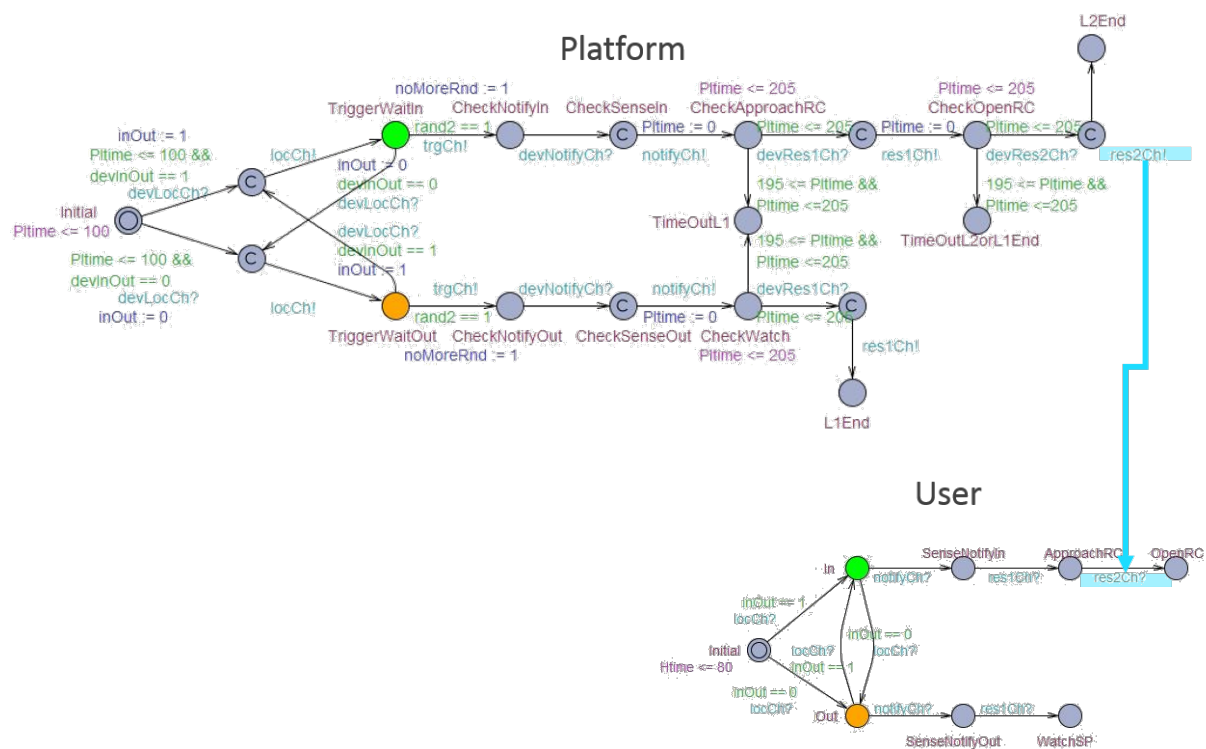


図 B.16: 現場応答完了の伝達

図 B.8、B.9、B.10、B.11、B.12、B.13、B.14、B.15、B.16 で見てきた状態遷移は成功パターンのみである。しかし、図 B.17 の四角で囲った部分のように、Ptime が 200 に達した場合はタイムアウトするような設計とした。これによって暗黙応答のみを行ったのか、現場応答まで行ったのかを判断することができるようになった。

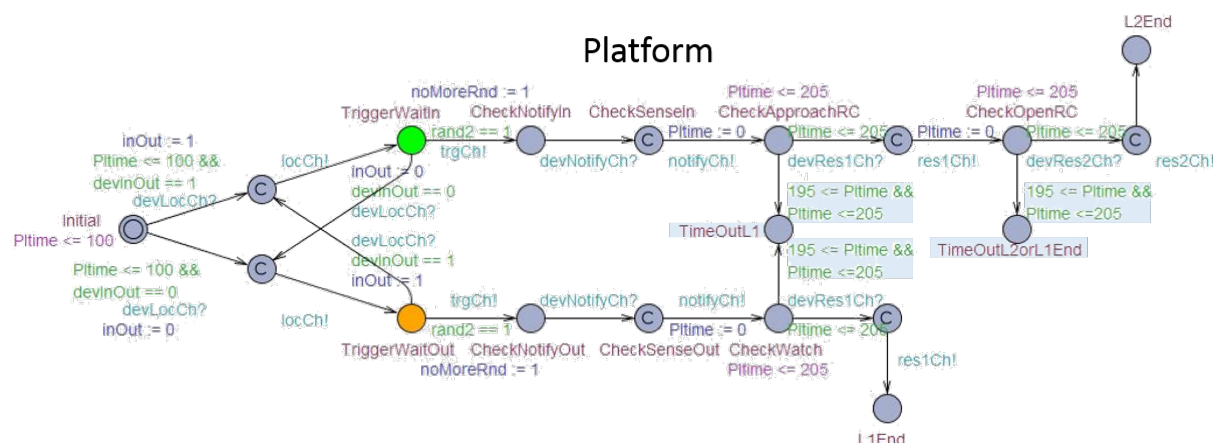


図 B.17: タイムアウトが発生した時の遷移

次ページより双方向通知応答基盤の実行結果ログを表示する。このログは屋内にいる時「炊飯器の炊き上がり時の動作完了通知」が起こり、ユーザが想定する行動を正しく取った場合の表示文字列である。赤文字が Device 系、黄文字が Resource 系、青文字が StateMachine 系である。なお、表示タイミングが前後している箇所があるので、注意されたい。


```

1. log4j:WARN No appenders could be found for logger (Jena).
2. log4j:WARN Please initialize the log4j system properly.
3. log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
4. 12月 26, 2018 8:42:26 午後 inr.DeviceManager loop
5. 情報: -----Loop program started-----
6. 12月 26, 2018 8:42:26 午後 inr.DeviceManager$1 run
7. 情報: Run: 1
8. -----RUN: 1 -----
9. 12月 26, 2018 8:42:26 午後 echowand.service.Core startService
10. 警告: subnet has already started
11. @prefix seas: <https://w3id.org/seas/> .
12. @prefix owl: <http://www.w3.org/2002/07/owl#> .
13. @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
14. @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
15. @prefix base_ontology:
    <https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
16. @prefix rdfs: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
17. @prefix inr: <http://ontology.jaist.ac.jp/kousuke/INR#> .
18. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
19. 情報: detectEOJs: 192.168.2.104: [0ef001, 000701, 03bb01, 02a001]
20. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
21. 情報: Add a new entry: Entry(192.168.2.104, 0ef001, Fresh)
22. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
23. 情報: Add a new entry: Entry(192.168.2.104, 000701, Fresh)
24. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
25. 情報: Add a new entry: Entry(192.168.2.104, 03bb01, Fresh)
26. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
27. 情報: Add a new entry: Entry(192.168.2.104, 02a001, Fresh)
28. 12月 26, 2018 8:42:27 午後 inr.DeviceManager$2 detectEOJsJoined
29. 情報: ----ECHONET Interface Monitoring Joined EOJ: 192.168.2.104 [0ef001, 000701, 03bb01,
    02a001]
30. 2018-12-26T20:42:27.32299Z
31.      Creating HumanDetectionSensor object from ECHONET frame...
32. 2018-12-26T20:42:27.32551Z
33.      Creating RiceCooker object from ECHONET frame...
34. 2018-12-26T20:42:27.47607Z
35.      Creating Buzzer object from ECHONET frame...
36. 2018-12-26T20:42:27.48130Z
37. 12月 26, 2018 8:42:27 午後 inr.DeviceManager$2 detectEOJsJoined
38. 情報: DeviceListSize = 1
39. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
40. 情報: detectEOJs: 192.168.2.101: [0ef001]
41. 12月 26, 2018 8:42:27 午後 jaist.tanlab.rm.monitor.Monitor detectEOJs
42. 情報: Add a new entry: Entry(192.168.2.101, 0ef001, Fresh)
43. 12月 26, 2018 8:42:27 午後 inr.DeviceManager$2 detectEOJsJoined
44. 情報: ----ECHONET Interface Monitoring Joined EOJ: 192.168.2.101 [0ef001]
45. 2018-12-26T20:42:27.48897Z
46. 12月 26, 2018 8:42:27 午後 inr.DeviceManager$2 detectEOJsJoined
47. 情報: DeviceListSize = 2
48. human detection status GET: exist
49. Node:192.168.2.104@EOJ:000701 {EPC:0x81, EDT: 0x19}=={InstallationLocation:Kitchen}
50. cover open/close status GET: closing
51. Node:192.168.2.104@EOJ:03bb01 {EPC:0x81, EDT: 0x19}=={InstallationLocation:Kitchen}
52. Sound generation setting GET: disable
53. soundGenerationSetting is buzzer disabled
54. Node:192.168.2.104@EOJ:02a001 {EPC:0x81, EDT: 0x19}=={InstallationLocation:Kitchen}
55. human detection status OBSERVE: not found
56. 12月 26, 2018 8:42:31 午後 inr.DeviceManager$1 run
57. 情報: Run: 2
58. -----RUN: 2 -----
59. @prefix seas: <https://w3id.org/seas/> .

```



```

60. @prefix owl:    <http://www.w3.org/2002/07/owl#> .
61. @prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
62. @prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
63. @prefix base_ontology:
    <https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
64. @prefix rdfs:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
65. @prefix inr:     <http://ontology.jaist.ac.jp/kousuke/INR#> .
66.
67. inr:Environment0 a inr:Environment ;
68.     inr:hasZone seas:Kitchen .
69.
70. base_ontology:Buzzer_192.168.2.104_1_2018-12-26T20:42:27.48130Z
71.     a base_ontology:Buzzer , base_ontology:Device ;
72.     inr:locatedIn inr:Environment0 ;
73.     base_ontology:hasOperationStatus
74.         "ON" ;
75.     base_ontology:hasSoundGenerationSetting
76.         "buzzer disabled" .
77.
78. seas:Kitchen a seas:Zone .
79.
80. base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-26T20:42:27.32551Z
81.     a base_ontology:HumanDetectionSensor , base_ontology:Device ;
82.     inr:locatedIn inr:Environment0 ;
83.     base_ontology:hasHumanDetectionStatus
84.         "not detected" ;
85.     base_ontology:hasOperationStatus
86.         "ON" .
87.
88. base_ontology:RiceCooker_192.168.2.104_1_2018-12-26T20:42:27.47607Z
89.     a base_ontology:RiceCooker , base_ontology:Device ;
90.     inr:locatedIn inr:Environment0 ;
91.     base_ontology:hasCoverOpenCloseStatus
92.         "cover close" ;
93.     base_ontology:hasOperationStatus
94.         "ON" .
95. Start app!
96. (NOW STATE : Smartphone.Initial) go LocSensing < auto
97. ENTER STATE : Smartphone.LocSensing
98. (NOW STATE : Smartphone.LocSensing) go LocSensingCommit < auto
99. ENTER STATE : Smartphone.LocSensingCommit
100. setDevInOut("devIn") < setDevInOut
101. (NOW STATE : Smartphone.LocSensingCommit) go Wait < auto
102. ENTER STATE : Smartphone.Wait
103. channelSendDevLoc("SRiceCooker") < channelSendDevLoc
104. channelReceiveDevLoc() < channelReceiveDevLoc
105. (NOW STATE : Platform.Initial) go InCommit < devLoc
106. setInOut("in") < setInOut
107. ENTER STATE : Platform.InCommit
108. (NOW STATE : Platform.InCommit) go TriggerWaitIn < auto
109. ENTER STATE : Platform.TriggerWaitIn
110. channelSendLoc("SHuman") < channelSendLoc
111. channelReceiveLoc() < channelReceiveLoc
112. (NOW STATE : Human.Initial) go In < locate
113. ENTER STATE : Human.In
114. ---locationSense! 0---
115. (NOW STATE : Buzzer.Initial) go Wait < activate
116. (NOW STATE : Smartphone.Wait) go LocSensing < sense
117. ENTER STATE : Buzzer.Wait
118. ENTER STATE : Smartphone.LocSensing
119. (NOW STATE : HumanDetectionSensor.Initial) go Wait < activate
120. (NOW STATE : Smartphone.LocSensing) go LocSensingCommit < auto
121. ENTER STATE : HumanDetectionSensor.Wait
122. ENTER STATE : Smartphone.LocSensingCommit

```

```

123.      (NOW STATE      : Smartphone.LocSensingCommit) go Wait < auto
124.      ENTER STATE     : Smartphone.Wait
125.      (NOW STATE      : OpenCloseSensor.Initial) go Wait < activate
126.      ENTER STATE     : OpenCloseSensor.WaitchannelSendDevLoc("SRiceCooker") <
channelSendDevLoc
127.
128.      channelReceiveDevLoc() < channelReceiveDevLoc
129.      (NOW STATE      : Platform.TriggerWaitIn) don't move < devLoc
130.      ENTER STATE     : Platform.TriggerWaitIn
131.      (NOW STATE      : Platform.TriggerWaitIn) don't move < auto
132.      ENTER STATE     : Platform.TriggerWaitIn
133.      channelSendLoc("SHuman") < channelSendLoc
134.      channelReceiveLoc() < channelReceiveLoc
135.      (NOW STATE      : Human.In) don't move < locate
136.      ENTER STATE     : Human.In
137.      -----RUN: 3 -----
138.      12 月 26, 2018 8:42:36 午後 inr.DeviceManager$1 run
139.      情報: Run: 3
140.      @prefix seas:    <https://w3id.org/seas/> .
141.      @prefix owl:   <http://www.w3.org/2002/07/owl#> .
142.      @prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
143.      @prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
144.      @prefix base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
145.      @prefix rdfs:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
146.      @prefix inr:     <http://ontology.jaist.ac.jp/kousuke/INR#> .
147.
148.      base_ontology:RiceCooker_192.168.2.104_1_2018-12-26T20:42:27.47607Z
149.      a                base_ontology:RiceCooker , base_ontology:Device ;
150.      inr:locatedIn    inr:Environment0 ;
151.      base_ontology:hasCoverOpenCloseStatus
152.      "cover close" ;
153.      base_ontology:hasOperationStatus
154.      "ON" .
155.
156.      inr:Response1 a    inr:Response ;
157.      inr:control    base_ontology:RiceCooker_192.168.2.104_1_2018-12-
26T20:42:27.47607Z ;
158.      inr:need        base_ontology:RiceCooker .
159.
160.      base_ontology:Buzzer_192.168.2.104_1_2018-12-26T20:42:27.48130Z
161.      a                base_ontology:Buzzer , base_ontology:Device ;
162.      inr:locatedIn    inr:Environment0 ;
163.      base_ontology:hasOperationStatus
164.      "ON" ;
165.      base_ontology:hasSoundGenerationSetting
166.      "buzzer disabled" .
167.
168.      inr:tapPopup a    inr:Task ;
169.      inr:hasTarget    base_ontology:Smartphone0 .
170.
171.      inr:Response0 a    inr:Response ;
172.      inr:control    base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-
26T20:42:27.32551Z ;
173.      inr:need        base_ontology:HumanDetectionSensor .
174.
175.      inr:taro a          inr:User ;
176.      inr:locatedIn    inr:Environment0 ;
177.      inr:useApp        inr:App16999 .
178.
179.      inr:open a          inr:Task ;
180.      inr:hasTarget    inr:RiceCooker .
181.
182.      inr:RiceCooker a    inr:AimObject .

```

```

183.
184.     inr:approach a      inr:Task ;
185.         inr:hasTarget inr:RiceCooker .
186.
187.     seas:Kitchen a seas:Zone .
188.
189.     inr:Complete a inr:Genre .
190.
191.     inr:ReadyRice a inr:Trigger .
192.
193.     inr:Notification0 a inr:Notification ;
194.         inr:control base_ontology:Buzzer_192.168.2.104_1_2018-12-
26T20:42:27.48130Z ;
195.         inr:need     base_ontology:Buzzer .
196.
197.     base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-26T20:42:27.32551Z
198.         a      base_ontology:HumanDetectionSensor , base_ontology:Device ;
199.         inr:locatedIn inr:Environment0 ;
200.         base_ontology:hasHumanDetectionStatus
201.             "not detected" ;
202.         base_ontology:hasOperationStatus
203.             "ON" .
204.
205.     inr:Linker0 a      inr:Linker ;
206.         inr:hasNotification inr:Notification0 ;
207.         inr:hasResponse     inr:Response1 , inr:Response0 .
208.
209.     inr:Environment0 a inr:Environment ;
210.         inr:hasZone seas:Kitchen .
211.
212.     inr:Goal0 a      inr:Goal ;
213.         inr:archived "not started" ;
214.         inr:hasAimObject inr:RiceCooker ;
215.         inr:hasGenre     inr:Complete ;
216.         inr:hasLinker    inr:Linker0 ;
217.         inr:hasTask      inr:tapPopup , inr:open , inr:approach ;
218.         inr:hasTrigger    inr:ReadyRice .
219.
220.     inr:App16999 a inr:App .
221.
222.     base_ontology:Smartphone0
223.         a      base_ontology:Smartphone , base_ontology:Device .
224.     alert!
225.     ---send trigger!---
226.     triggerIn() < triggerIn
227.     (NOW STATE : Platform.TriggerWaitIn) go CheckNotifyIn < notifyTrigger
228.     ENTER STATE : Platform.CheckNotifyIn
229.     channelSendTrigger("SBuzzer") < channelSendTrigger
230.     channelReceiveTrigger() < channelReceiveTrigger
231.     (NOW STATE : Buzzer.Wait) go Commit < soundStart
232.     ENTER STATE : Buzzer.Commit
233.     (NOW STATE : Buzzer.Commit) go Notify < auto
234.     ENTER STATE : Buzzer.Notify
235.     ---soundOn---
236.     changeRDFtoE
237.     set Data!
238.     value is buzzer enabled ENABLE=buzzer enabledDISABLE=buzzer disabled
239.     Sound generation setting SET: enabled
240.     channelSendDevNotify("SRiceCooker") < channelSendDevNotify
241.     channelReceiveDevNotify() < channelReceiveDevNotify
242.     (NOW STATE : Platform.CheckNotifyIn) go CheckSenseIn < devNotify
243.     ENTER STATE : Platform.CheckSenseIn
244.     (NOW STATE : Platform.CheckSenseIn) go CheckApproachRC < auto
245.     ENTER STATE : Platform.CheckApproachRC

```

```

246.     channelSendNotify("SHuman") < channelSendNotify
247.     channelReceiveNotify() < channelReceiveNotify
248.     (NOW STATE      : Human.In) go SenseNotificationIn < senseNotification
249.     ENTER STATE      : Human.SenseNotificationIn
250.     channelSendTrigger("SHumanDetectionSensor") < channelSendTrigger
251.     channelReceiveTrigger() < channelReceiveTrigger
252.     (NOW STATE      : HumanDetectionSensor.Wait) go Sensing < humanSearch
253.     ENTER STATE      : HumanDetectionSensor.Sensing
254.     channelSendTrigger("SOpenCloseSensor") < channelSendTrigger
255.     channelReceiveTrigger() < channelReceiveTrigger
256.     (NOW STATE      : OpenCloseSensor.Wait) go Sensing < openTrack
257.     ENTER STATE      : OpenCloseSensor.Sensing
258.     PREFIX :      <http://ontology.jaist.ac.jp/kousuke/INR#>
259.     PREFIX inr:    <http://ontology.jaist.ac.jp/kousuke/INR#>
260.     PREFIX base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#>
261.     PREFIX seas:  <https://w3id.org/seas/>
262.     PREFIX owl:  <http://www.w3.org/2002/07/owl#>
263.     PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
264.     PREFIX rdfs:   <http://www.w3.org/2000/01/rdf-schema#>
265.     PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
266.     Delete {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
267.     WHERE
268.     {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
269.     Sound generation setting OBSERVE: enable
270.     soundGenerationSetting is buzzer enabled
271.     human detection status GET: not exist
272.     cover open/close status GET: closing
273.     Sound generation setting GET: enable
274.     soundGenerationSetting is buzzer enabled
275.     human detection status OBSERVE: found
276.     12 月 26, 2018 8:42:41 午後 inr.DeviceManager$1 run
277.     情報: Run: 4
278.     -----RUN: 4 -----
279.     @prefix seas:  <https://w3id.org/seas/> .
280.     @prefix owl:  <http://www.w3.org/2002/07/owl#> .
281.     @prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
282.     @prefix xsd:    <http://www.w3.org/2001/XMLSchema#> .
283.     @prefix base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
284.     @prefix rdfs:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
285.     @prefix inr:    <http://ontology.jaist.ac.jp/kousuke/INR#> .
286.
287.     base_ontology:RiceCooker_192.168.2.104_1_2018-12-26T20:42:27.47607Z
288.     a                    base_ontology:RiceCooker , base_ontology:Device ;
289.     inr:locatedIn inr:Environment0 ;
290.     base_ontology:hasCoverOpenCloseStatus
291.     "cover close" ;
292.     base_ontology:hasOperationStatus
293.     "ON" .
294.
295.     inr:Response1 a      inr:Response ;
296.     inr:control   base_ontology:RiceCooker_192.168.2.104_1_2018-12-
26T20:42:27.47607Z ;
297.     inr:need      base_ontology:RiceCooker .
298.
299.     base_ontology:Buzzer_192.168.2.104_1_2018-12-26T20:42:27.48130Z
300.     a                    base_ontology:Buzzer , base_ontology:Device ;
301.     inr:locatedIn inr:Environment0 ;
302.     base_ontology:hasOperationStatus
303.     "ON" ;
304.     base_ontology:hasSoundGenerationSetting

```

```

305.                "buzzer enabled" .
306.
307.    inr:tapPopup a      inr:Task ;
308.        inr:hasTarget base_ontology:Smartphone0 .
309.
310.    inr:Response0 a      inr:Response ;
311.        inr:control base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-
26T20:42:27.32551Z ;
312.        inr:need      base_ontology:HumanDetectionSensor .
313.
314.    inr:taro a      inr:User ;
315.        inr:locatedIn inr:Environment0 ;
316.        inr:useApp    inr:App16999 .
317.
318.    inr:open a      inr:Task ;
319.        inr:hasTarget inr:RiceCooker .
320.
321.    inr:RiceCooker a inr:AimObject .
322.
323.    inr:approach a      inr:Task ;
324.        inr:hasTarget inr:RiceCooker .
325.
326.    seas:Kitchen a seas:Zone .
327.
328.    inr:Complete a inr:Genre .
329.
330.    inr:ReadyRice a inr:Trigger .
331.
332.    inr:Notification0 a inr:Notification ;
333.        inr:control base_ontology:Buzzer_192.168.2.104_1_2018-12-
26T20:42:27.48130Z ;
334.        inr:need      base_ontology:Buzzer .
335.
336.    base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-26T20:42:27.32551Z
337.        a      base_ontology:HumanDetectionSensor , base_ontology:Device ;
338.        inr:locatedIn inr:Environment0 ;
339.        base_ontology:hasHumanDetectionStatus
340.            "detected" ;
341.        base_ontology:hasOperationStatus
342.            "ON" .
343.
344.    inr:Linker0 a      inr:Linker ;
345.        inr:hasNotification inr:Notification0 ;
346.        inr:hasResponse    inr:Response1 , inr:Response0 .
347.
348.    inr:Environment0 a inr:Environment ;
349.        inr:hasZone seas:Kitchen .
350.
351.    inr:Goal0 a      inr:Goal ;
352.        inr:archived "wait response" ;
353.        inr:hasAimObject inr:RiceCooker ;
354.        inr:hasGenre      inr:Complete ;
355.        inr:hasLinker      inr:Linker0 ;
356.        inr:hasTask        inr:tapPopup , inr:open , inr:approach ;
357.        inr:hasTrigger      inr:ReadyRice .
358.
359.    inr:App16999 a inr:App .
360.
361.    base_ontology:Smartphone0
362.        a      base_ontology:Smartphone , base_ontology:Device .
363.    cover open/close status OBSERVE: opened
364.    (NOW STATE      : HumanDetectionSensor.Sensing) go Detect < detect
365.    ENTER STATE      : HumanDetectionSensor.Detect
366.    channelSendDevRes1("SRiceCooker") < channelSendDevRes1

```

```

367.     channelReceiveDevRes1() < channelReceiveDevRes1
368.     (NOW STATE      : Platform.CheckApproachRC) go Res1InCommit < devRes1
369.     ENTER STATE     : Platform.Res1InCommit
370.     (NOW STATE      : Platform.Res1InCommit) go CheckOpenRC < auto
371.     ENTER STATE     : Platform.CheckOpenRC
372.     channelSendRes1("SHuman") < channelSendRes1
373.     channelReceiveRes1() < channelReceiveRes1
374.     (NOW STATE      : Human.SenseNotificationIn) go ApproachRC < res1
375.     ENTER STATE     : Human.ApproachRC
376.     PREFIX :      <http://ontology.jaist.ac.jp/kousuke/INR#>
377.     PREFIX inr:    <http://ontology.jaist.ac.jp/kousuke/INR#>
378.     PREFIX base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#>
379.     PREFIX seas:   <https://w3id.org/seas/>
380.     PREFIX owl:   <http://www.w3.org/2002/07/owl#>
381.     PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
382.     PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
383.     PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
384.     Delete {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
385.     WHERE
386.     {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
387.     human detection status OBSERVE: not found
388.     ---locationSense! 1---ENTER STATE      : Smartphone.LocSensing
389.     (NOW STATE      : Smartphone.Wait) go LocSensing < sense
390.     (NOW STATE      : Smartphone.LocSensing) go LocSensingCommit < auto
391.     (NOW STATE      : Smartphone.LocSensingCommit) go Wait < auto
392.     channelSendDevLoc("SRiceCooker") < channelSendDevLoc
393.     channelReceiveDevLoc() < channelReceiveDevLoc
394.     (NOW STATE      : Platform.CheckOpenRC) don't move < devLoc
395.     (NOW STATE      : Platform.CheckOpenRC) don't move < auto
396.     channelSendLoc("SHuman") < channelSendLoc
397.     channelReceiveLoc() < channelReceiveLoc
398.     (NOW STATE      : Human.ApproachRC) don't move < locate
399.
400.     ENTER STATE     : Smartphone.LocSensingCommit
401.     ENTER STATE     : Smartphone.Wait
402.     ENTER STATE     : Platform.CheckOpenRC
403.     ENTER STATE     : Platform.CheckOpenRC
404.     ENTER STATE     : Human.ApproachRC
405.     12 月 26, 2018 8:42:46 午後 inr.DeviceManager$1 run
406.     情報: Run: 5
407.     -----RUN: 5 -----
408.     @prefix seas:   <https://w3id.org/seas/> .
409.     @prefix owl:   <http://www.w3.org/2002/07/owl#> .
410.     @prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
411.     @prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
412.     @prefix base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
413.     @prefix rdfs:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
414.     @prefix inr:     <http://ontology.jaist.ac.jp/kousuke/INR#> .
415.
416.     base_ontology:RiceCooker_192.168.2.104_1_2018-12-26T20:42:27.47607Z
417.         a                               base_ontology:RiceCooker , base_ontology:Device ;
418.         inr:locatedIn inr:Environment0 ;
419.         base_ontology:hasCoverOpenCloseStatus
420.             "cover open" ;
421.         base_ontology:hasOperationStatus
422.             "ON" .
423.
424.     inr:Response1 a inr:Response ;
425.     inr:control   base_ontology:RiceCooker_192.168.2.104_1_2018-12-
26T20:42:27.47607Z ;

```

```

426.         inr:need      base_ontology:RiceCooker .
427.
428.     base_ontology:Buzzer_192.168.2.104_1_2018-12-26T20:42:27.48130Z
429.         a              base_ontology:Buzzer , base_ontology:Device ;
430.         inr:locatedIn  inr:Environment0 ;
431.         base_ontology:hasOperationStatus
432.             "ON" ;
433.         base_ontology:hasSoundGenerationSetting
434.             "buzzer enabled" .
435.
436.     inr:tapPopup a      inr:Task ;
437.         inr:hasTarget  base_ontology:Smartphone0 .
438.
439.     inr:Response0 a     inr:Response ;
440.         inr:control    base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-
26T20:42:27.32551Z ;
441.         inr:need      base_ontology:HumanDetectionSensor .
442.
443.     inr:taro a          inr:User ;
444.         inr:locatedIn  inr:Environment0 ;
445.         inr:useApp     inr:App16999 .
446.
447.     inr:open a          inr:Task ;
448.         inr:hasTarget  inr:RiceCooker .
449.
450.     inr:RiceCooker a   inr:AimObject .
451.
452.     inr:approach a     inr:Task ;
453.         inr:hasTarget  inr:RiceCooker .
454.
455.     seas:Kitchen a     seas:Zone .
456.
457.     inr:Complete a     inr:Genre .
458.
459.     inr:ReadyRice a    inr:Trigger .
460.
461.     inr:Notification0 a inr:Notification ;
462.         inr:control    base_ontology:Buzzer_192.168.2.104_1_2018-12-
26T20:42:27.48130Z ;
463.         inr:need      base_ontology:Buzzer .
464.
465.     base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-26T20:42:27.32551Z
466.         a              base_ontology:HumanDetectionSensor , base_ontology:Device ;
467.         inr:locatedIn  inr:Environment0 ;
468.         base_ontology:hasHumanDetectionStatus
469.             "not detected" ;
470.         base_ontology:hasOperationStatus
471.             "ON" .
472.
473.     inr:Linker0 a       inr:Linker ;
474.         inr:hasNotification inr:Notification0 ;
475.         inr:hasResponse    inr:Response1 , inr:Response0 .
476.
477.     inr:Environment0 a  inr:Environment ;
478.         inr:hasZone     seas:Kitchen .
479.
480.     inr:Goal0 a         inr:Goal ;
481.         inr:archived     "L1" ;
482.         inr:hasAimObject inr:RiceCooker ;
483.         inr:hasGenre     inr:Complete ;
484.         inr:hasLinker    inr:Linker0 ;
485.         inr:hasTask      inr:tapPopup , inr:open , inr:approach ;
486.         inr:hasTrigger   inr:ReadyRice .
487.

```

```

488.      inr:App16999 a inr:App .
489.
490.      base_ontology:Smartphone0
491.          a      base_ontology:Smartphone , base_ontology:Device .
492.      ---soundOff---
493.      changeRDFtoE
494.      set Data!
495.      value is buzzer disabled ENABLE=buzzer enabledDISABLE=buzzer disabled
496.      Sound generation setting SET: disabled
497.      (NOW STATE      : Buzzer.Notify) go End < soundEnd
498.      ENTER STATE      : Buzzer.End
499.      Sound generation setting OBSERVE: disable
500.      soundGenerationSetting is buzzer disabled
501.      human detection status GET: not exist
502.      cover open/close status GET: opening
503.      Sound generation setting GET: disable
504.      soundGenerationSetting is buzzer disabled
505.      human detection status OBSERVE: found
506.      >>>opened() < opened
507.      ENTER STATE      : OpenCloseSensor.Open
508.      (NOW STATE      : OpenCloseSensor.Sensing) go Open < open
509.      channelSendDevRes2("SRiceCooker") < channelSendDevRes2
510.      channelReceiveDevRes2() < channelReceiveDevRes2
511.      (NOW STATE      : Platform.CheckOpenRC) go Res2InCommit < devRes2
512.      (NOW STATE      : Platform.Res2InCommit) go L2End < auto
513.      channelSendRes2("SHuman") < channelSendRes2
514.      channelReceiveRes2() < channelReceiveRes2
515.      (NOW STATE      : Human.ApproachRC) go OpenRC < res2
516.      PREFIX :      <http://ontology.jaist.ac.jp/kousuke/INR#>
517.      PREFIX inr:    <http://ontology.jaist.ac.jp/kousuke/INR#>
518.      PREFIX base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#>
519.      PREFIX seas:   <https://w3id.org/seas/>
520.      PREFIX owl:   <http://www.w3.org/2002/07/owl#>
521.      PREFIX rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
522.      PREFIX rdfs:    <http://www.w3.org/2000/01/rdf-schema#>
523.      PREFIX xsd:     <http://www.w3.org/2001/XMLSchema#>
524.      Delete {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
525.      WHERE
526.      {<http://ontology.jaist.ac.jp/kousuke/INR#Goal0>
<http://ontology.jaist.ac.jp/kousuke/INR#archived> ?any}
527.      ENTER STATE      : Platform.Res2InCommit
528.      ENTER STATE      : Platform.L2End
529.      ENTER STATE      : Human.OpenRC
530.      ---In L2 End...---
531.      cover open/close status OBSERVE: closed
532.      12 月 26, 2018 8:42:51 午後 inr.DeviceManager$1 run
533.      情報: Run: 6
534.      -----RUN: 6 -----
535.      @prefix seas:   <https://w3id.org/seas/> .
536.      @prefix owl:   <http://www.w3.org/2002/07/owl#> .
537.      @prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
538.      @prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .
539.      @prefix base_ontology:
<https://git.onem2m.org/MAS/BaseOntology/raw/master/base_ontology.owl#> .
540.      @prefix rdfs:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
541.      @prefix inr:     <http://ontology.jaist.ac.jp/kousuke/INR#> .
542.
543.      base_ontology:RiceCooker_192.168.2.104_1_2018-12-26T20:42:27.47607Z
544.          a      base_ontology:RiceCooker , base_ontology:Device ;
545.          inr:locatedIn inr:Environment0 ;
546.          base_ontology:hasCoverOpenCloseStatus
547.              "cover close" ;

```



```

548.          base_ontology:hasOperationStatus
549.              "ON" .
550.
551.      inr:Response1 a      inr:Response ;
552.      inr:control base_ontology:RiceCooker_192.168.2.104_1_2018-12-
26T20:42:27.47607Z ;
553.      inr:need      base_ontology:RiceCooker .
554.
555.      base_ontology:Buzzer_192.168.2.104_1_2018-12-26T20:42:27.48130Z
556.      a      base_ontology:Buzzer , base_ontology:Device ;
557.      inr:locatedIn inr:Environment0 ;
558.      base_ontology:hasOperationStatus
559.          "ON" ;
560.      base_ontology:hasSoundGenerationSetting
561.          "buzzer disabled" .
562.
563.      inr:tapPopup a      inr:Task ;
564.      inr:hasTarget base_ontology:Smartphone0 .
565.
566.      inr:Response0 a      inr:Response ;
567.      inr:control base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-
26T20:42:27.32551Z ;
568.      inr:need      base_ontology:HumanDetectionSensor .
569.
570.      inr:taro a      inr:User ;
571.      inr:locatedIn inr:Environment0 ;
572.      inr:useApp      inr:App16999 .
573.
574.      inr:open a      inr:Task ;
575.      inr:hasTarget inr:RiceCooker .
576.
577.      inr:RiceCooker a inr:AimObject .
578.
579.      inr:approach a      inr:Task ;
580.      inr:hasTarget inr:RiceCooker .
581.
582.      seas:Kitchen a seas:Zone .
583.
584.      inr:Complete a inr:Genre .
585.
586.      inr:ReadyRice a inr:Trigger .
587.
588.      inr:Notification0 a inr:Notification ;
589.      inr:control base_ontology:Buzzer_192.168.2.104_1_2018-12-
26T20:42:27.48130Z ;
590.      inr:need      base_ontology:Buzzer .
591.
592.      base_ontology:HumanDetectionSensor_192.168.2.104_1_2018-12-26T20:42:27.32551Z
593.      a      base_ontology:HumanDetectionSensor , base_ontology:Device ;
594.      inr:locatedIn inr:Environment0 ;
595.      base_ontology:hasHumanDetectionStatus
596.          "detected" ;
597.      base_ontology:hasOperationStatus
598.          "ON" .
599.
600.      inr:Linker0 a      inr:Linker ;
601.      inr:hasNotification inr:Notification0 ;
602.      inr:hasResponse      inr:Response1 , inr:Response0 .
603.
604.      inr:Environment0 a inr:Environment ;
605.      inr:hasZone seas:Kitchen .
606.
607.      inr:Goal0 a      inr:Goal ;
608.      inr:archived      "L2" ;

```

```

609.          inr:hasAimObject  inr:RiceCooker ;
610.          inr:hasGenre      inr:Complete ;
611.          inr:hasLinker     inr:Linker0 ;
612.          inr:hasTask       inr:tapPopup , inr:open , inr:approach ;
613.          inr:hasTrigger    inr:ReadyRice .
614.
615.  inr:App16999  a  inr:App .
616.
617.  base_ontology:Smartphone0
618.      a          base_ontology:Smartphone , base_ontology:Device .
619.  (NOW STATE    : SHumanDetectionSensor.Default?) go  Error < detect
620.  ENTER STATE   : HumanDetectionSensor.Error
621.  ENTER STATE   : Platform.L2End
622.  ENTER STATE   : Platform.L2End

```