情報処理学会論文誌 Vol.58 No.10 1591–1605 (Oct. 2017)

物語記述による ホームネットワークアプリケーション開発手法の提案

大村 廉[1,a)]

受付日 2016年12月23日, 採録日 2017年7月4日

概要：ホームネットワーク上のアプリケーションの作成は，カスタマイズ性やメンテナンス性の観点から，その家の住人自身が開発を行うことが望ましい．提案手法では，ホームネットワーク上の各デバイスを登場人物としてとらえ，その動作を ECA ルールなどのルールに基づく簡易な自然言語（日本語）で定義できるようにする．また，デバイス間の通信をデバイスどうしの自由な会話によって定義できるようにする．さらに，接続詞を用いて冗長な表現を排除し，文書としての自然さを損なわずにデバイス連携の記述が行えるようにする．これにより，提案手法に基づくアプリケーション開発環境「PlayHouse」をタブレット端末上に実装し，こども科学館でのイベントで展示を行った．被験者には十数分から数十分間，自由にアプリケーション作成を行って貰った後，アンケートに回答してもらった．この結果，10 歳以下の子供には漢字の判読やタブレット上の文字入力について難しさが残るものの，小学校高学年以上は提案手法によって簡単に，かつ，楽しみながらアプリケーションを作成することができることを確認し，本手法の有効性を確認することができた．

キーワード：ホームネットワーク，アプリケーション開発環境，物語記述アプローチ，デバイス擬人化

Story Creation Approach for Home Network Application Development

Ren Ohumura[1,a)]

Received: December 23, 2016, Accepted: July 4, 2017

Abstract: A home network application ought to be built by a resident from the view point of its cutamizability and maintainanace. In this paper, story creation approach is proposed as a development method of home network aplications with the aim to let kids and housewives, who generally stay longer home, build their own applications. In our method, each device is dealt as a charactor of a story, and its operations are defined by simple natural language corresponding with a simple rule, such as ECA. Communication between devices for their cooperation is defined as a free conversation between them. Conjunction words are used for eliminating redundant expression, which is required in direct rule writing, so that the written sentences keep natural. “PlayHouse” was implemented on a tablet PC as a prototype of our method, and exhibited it at an event in a kids science museum. In the experiments, subjects created their own applications after getting explaination of the basic usage and were asked to answer our questionarie. The results showed that kids over 10 years old could easily create some applications with fun, and effectiveness of our proposed method is confirmed, although kids under or equal to 10 years old suffered from reading chinese characters and inputting text on a tablet PC.

Keywords: home network, application development environment, story creation approach, device personification

1. はじめに

組込技術やネットワーク技術の進歩により，一般の家庭にもセンシング能力，計算能力，通信能力を持った製品（デバイス）が普及してきている．そして，ホームネットワー

[1] 豊橋技術科学大学
Toyohashi University of Technology, Toyohashi, Aichi 441–8580, Japan
[a)] ren@tut.jp

© 2017 Information Processing Society of Japan

クを構築して各デバイスを連携させ，人の生活を支援する技術の開発が進められている [1], [2], [3], [4], [5]．しかし，デバイスを接続してホームネットワークが構築できたとしても，それぞれの家の大きさや部屋の間取りなどは異なり，使用されるデバイスの種類や数，設置位置は家庭ごとに違いが生じる．また，家庭ごとに家族構成や生活習慣は異なり，ホームネットワークに要求する支援の内容にもそれぞれ違いが生じる．すなわち，ホームネットワークのアプリケーションはそれぞれの家庭の環境や要求の違いに応じて作成したり，カスタマイズされる必要がある．

アプリケーションの作成やカスタマイズを「誰が行うか」という点について，大きく 2 つの方針を考えることができる．1 つは，機器のメーカや専門業者が行う方針であり，もう 1 つは家の住人が自身で行う方針である．前者の場合，専門的な技能を持つ者が各家庭で要求するサービス内容を住人からヒヤリングし，それをもとにアプリケーションを作成することになる．複雑な制御をともなう高度なアプリケーションが作成可能となるが，反面，住人の要求がうまく伝達されず，結局住人にとって本当に使いやすいアプリケーションが構築されなくなる可能性も高まる．また，軽微な修正を行う場合も，その修正に長い時間や多大なコストがかかることになる．結果として，生活の変化などに対応することも難しくなってしまう．一方，後者のよう，家の住人自身で行う場合は，複雑な制御をともなうアプリケーションの構築は難しいものの，それぞれの生活形態に合わせた住人にとって使い心地の良いアプリケーションが構築できる可能性が高まる．また，軽微な修正を住人自身で行い，細かい調整や生活形態に合わせてアプリケーションを修正していく，といったことも可能になる．このことは文献 [6] でも指摘されており，加えて，住人達が創意工夫を凝らした新しいサービスを考えることができるようにするためにも，ホームネットワークにおけるアプリケーションは住人達が自分自身で作成できるようにする必要があることが指摘されている．

また，文献 [6] や文献 [7] では，ホームネットワークアプリケーションは，重厚で長大・巨大なアプリケーションよりも軽くて単機能なアプリケーションが好まれることが指摘されている．さらに，文献 [8] では，ユーザから収集したアプリケーション例のうち，約 78%が「○○のときに，○○をして欲しい」といった単一の Trigger-Action ルールで記述可能であることが指摘されている．さらに，文献 [7] では，住人がアプリケーション開発を行う場合アプリケーション開発自体が「楽しい」作業と感じられる必要があることが指摘されている．一般的な家庭を考えた場合，最も長く家に滞在するのは主婦/主夫，子供であると考えられる．すなわち，ホームネットワークにおけるアプリケーション開発では，簡易な動作ルールを，主婦/主夫や子供が楽しみながら自身のアイデアに基づき開発を行える環境であることが望ましい．

そこで本研究では，情報技術にあまりなじみのない人や子供などであっても，ホームネットワークアプリケーションを作成できるようにするため，各デバイスの動作や連携を「物語（おはなし）をつくる」ように記述してプログラミングを行う手法を提案する．そして，その手法に基づくアプリケーション開発環境「PlayHouse」を実現した．架空の登場人物などを設定し，1 人，あるいは両親や友達と「おはなし」を作る，という作業は誰もが幼少期には「遊び」の 1 つとして行ったことであると考えられる．すなわち，本研究では，ホームネットワークアプリケーションの開発を物語記述のアプローチを用いて「遊び」に落とし込み，ユーザが「容易に」かつ「楽しさ」を感じながらアプリケーション開発を行える環境を提供する手法を提案する．

デバイス連携によるホームネットワークアプリケーションの開発には，個々のデバイスの動作と同時に，デバイス間の連携，および，そのために取り交わされるメッセージの定義が重要となる．提案手法では，デバイスを物語における登場人物としてとらえ，デバイスの動作は登場人物の様子や動作として自然言語（日本語）を用いて定義できるようにする．デバイスの動作は簡易な Trigger-Action ルール [8] や ECA（Event-Condition-Action）ルール [9], [10] で定義されるものとし，Trigger（Event）や Condition，Action に対して，あらかじめ定義された自然言語の表現から選択を行うようにする．これにより，動作記述の曖昧性を排除する．また，メッセージは「デバイス間で取り交わされる会話」としてやはり自然言語で定義できるようにする．このメッセージは制限なく自由な記述ができるようにすることで物語としての自由度を確保する．さらに，Trigger-Action ルールや ECA ルールにおいて多段のデバイス連携を行うには，あるデバイスの「イベントとなる動作（Action）」を定義するとともに，連携する別のデバイスについて「その動作を観測したとき（Trigger もしくは Action）」の内容を各段において記述する必要がある．これは冗長な記述となり，文章としての不自然さや読みにくさを招く．このため，本研究では接続詞を用いてこの冗長性を排除して文章の自然さを確保する．

提案手法の評価のため，これらの手法を実装したホームネットワークアプリケーション開発環境 PlayHouse をタブレット PC 上に実装した．そして，数種類のホームネットワーク機器とともに愛知県豊橋市のこども未来館におけるイベントにて展示を行った．来場者にそのアプリケーション開発環境を用いて自由にホームネットワークアプリケーションの作成を行ってもらい，アンケートに答えてもらって提案手法の有効性の確認を行った．

以下，2 章で関連研究について述べる．3 章で本研究が想定するホームネットワーク環境について概説した後，4 章で提案手法を詳しく説明する．5 章で，提案手法を実装し

© 2017 Information Processing Society of Japan

たアプリケーション開発環境について述べ，6 章で，アンケート結果の報告と提案手法の有効性に関する考察を行う．7 章で，本稿をまとめる．

2. 関連研究

センサネットワークやホームネットワーク上のアプリケーションプログラミングを容易にし，プログラミングに不慣れな人でもデバイスどうしの連携に基づくアプリケーション開発を行えるようにすることを目指した研究は多く存在している．これらの研究では主な方針として，動作定義に必要なプログラミング記述をスプリクト化してデバイスの動作・連携の記述を簡易にする方法 [11]，プログラミング記述をなくしボタン操作でアプリケーション開発を可能にする方法 [12]，デバイスの機能やサービスが設定されたアイコンを線で繋ぎ組み合わせていくことでデバイスどうしを連携させる方法（ビジュアルプログラミング）[13]，などが提案されている．しかし，これらの方法は依然として変数の扱いやフロー制御などを意識する必要があり，プログラミングに対する知識を要求するものが多い．本研究でターゲットととする子供などは，このような情報技術に親しみが薄い場合が多く，これらの既存手法では依然として困難が生じると考えられる．

現在の計算機プログラムの多くは，人工的に開発された言語で作成される．これに対して，主に初心者に対するプログラミングの敷居の高さを低減することを目的とし，自然言語によるプログラムや，使用する単語を日常使われる単語に類似させて自然言語に近い形式言語でプログラミングを行うプログラミング言語も多く提案されてきた．たとえば，日本語によるプログラム言語として「和漢」[14] や「言霊」[15] などの研究や，日本語 Mind [16]，なでしこ [17] などが存在する．教育用プログラム開発環境である Scratch [18] も日本語によるアプリケーション開発ができるようになっている．本研究もこれらの研究と同様に，日本語の単語を用いてデバイスの動作の定義を行い，情報技術に馴染みがないユーザへの敷居を低減させるようにする．しかし，既存の日本語によるプログラミング言語では，主として 1 つの計算機上で動作するプログラムの動作定義を行うことを対象としたものであった．これに対し，本研究では各デバイスの動作とともにデバイス間の連携がその定義の主な対象となる．そして，デバイスを擬人化するとともに，デバイス間のやりとりをデバイスどうしの会話として成立させることで，ネットワークで結合された分散システムとして存在する各デバイス上で，連携して実行されるルールの連続を，ストーリーとして定義可能とするものである．そして，「自由に物語を作成する」という，1 つの「遊び」としてホームネットワークアプリケーションの開発を位置づける．

IFTTT [19] は元来 Web サービスどうしを Trigger-Action ルールによって連携させることを目的としていた．最近はその対象を IoT デバイスに広げ，たとえばドアやライト，電源プラグなどの連携が定義できるようになっている．IFTTT では Trigger-Action によるサービスの連携を “If this then that” における this と then の部分を一覧から選択することで定義する．また，よく利用される連携の定義はあらかじめシステムに登録されており，ユーザはこれらを選択して利用することができる．文献 [8] では，IFTTT を用いたスマートホームアプリケーションの種類や実現可能性について分析を行い，その適用範囲やインタフェースの容易性などを示した．しかし，ルールの定義は 1 段の連携のみであり，1 つの Trigger に対する Action の発火が，別のデバイスの Action を引き起こすといった多段の連携には，複数のルールを別々に記述する必要がある．このため，連続するデバイスの動作について一連の流れを把握しながら定義を行うことは難しい．また，ルールの設定においても，無機質に記述された動作の一覧から選ぶものであり，また，連携も非明示的に行われる．一方，提案手法は「おはなし」として，動作の流れを意識しながら多段の連携を記述することができる．また，デバイスの動作は「おはなし」のキャラクタの動きとして動作定義が行われ，デバイス間のやりとりもそれらのキャラクタの会話として位置づけられる．このため，作成した動作定義に対して，親しみや楽しさを与える要素がより多く，特に情報技術に不慣れなユーザには，アプリケーション開発に対する敷居を下げることができると考えられる．

3. 想定環境

本研究では，家庭などの住環境において各種デバイスが接続されたネットワーク（ホームネットワーク）を想定する．ホームネットワークに接続されるデバイスとしては主に家電（エアコン，冷蔵庫，ポット，照明，掃除機，時計，テレビ，オーディオなど，所謂，白物家電や黒物家電）やセンサ（温度計，ドアの開閉センサや焦電センサなどの人感センサ）などを想定する．各デバイスにはプロセッサが付与され，計算能力（プログラム実行能力）を持つものとする．前述のよう，各デバイスの動作は Trigger-Action ルール，あるいは，ECA ルールによって定義されるものとする．たとえば，テレビであれば，内蔵のタイマを用いて「○時になったら」という Trigger（Event）を契機に「電源の ON/OFF をする」，や「チャンネルを○に切り替える」などといった Action を行うことが可能であるものとする．また，ネットワークを介して他のデバイスとのメッセージのやりとりが行えるものとする．そして，デバイスは他のデバイスからある特定のメッセージを受信したことを，ルール実行のトリガ Trriger（Event）として定義できるものとする．同様に，それぞれのデバイスは他のデバイスにある特定のメッセージを送るということをその Action

© 2017 Information Processing Society of Japan

として定義できるものとする．すなわち，個々のデバイスはある特定のメッセージの受信を機にルールを実行させることができるとともに，Action としてある特定のメッセージを送信できるものとする．

このような条件を満たすアプリケーション実行環境として，我々は ECA ルールと Twitter を用いたアプリケーション実行環境の開発を行っている [20]．このアプリケーション実行環境では，ルールとしてルール発動の契機となるイベント（Event），その際に評価される条件（Condition），および，その時の動作（Action）からなる ECA ルール [21] に基づくルールの記述を可能としている．また，メッセージシステムとして Twitter を用いている．ホームネットワーク上の各デバイスは相互にフォロー関係にある，という前提があるものの，Twitter を用いてデバイスどうしは相互にメッセージのやりとりを行うことができる．さらに，Twitter から特定のメッセージを受信した時を ECA ルールの Event として定義できるとともに，デバイスの Action として Twitter へメッセージが送信できるようにしている．なお，文献 [20] の研究でも，上記のようなアプリケーション実行基盤を提案するとともに，開発環境の実装を行っている．この開発環境においても，ストーリーとしての演出を志向したが，ボタンベースのインタフェースでの実装であり，デバイスの擬人化やデバイスどうしの会話について積極的な演出は行われていなかった．本研究では，このような実行環境を前提とし，物語記述アプローチによるアプリケーション開発手法を提案する．

4. 提案手法

本研究では，主な使用者として情報技術に親しみの薄い子供などを想定し，そのような子供らでも，容易，かつ，楽しみながらアプリケーションの開発が行える環境を構築することを目的とする．その手法として「物語（おはなし）を作成するように開発を行う」方法を提案し，その方法を実装した開発環境を構築する．

これを実現するための具体的な方法として，

- 「おはなし」の登場人物として各デバイスを擬人化
- 日本語をベースとしたデバイスの動作記述
- 自由な会話内容の設定
- 接続詞による冗長な記述の回避

を行う．

以下，それぞれの項目について説明する．

4.1 記述例とデバイスの擬人化

まず，提案手法の概要を説明するため，例として「室温が 10 度になったとき，ストーブをつける」というアプリケーションを考える．ただし，室温を検知する温度計は部屋に設置してある温度計であり，ストーブとは別に設置されているものであるとする．すなわち，温度計が 10 度を検知したときにメッセージを送り，ストーブを動作させる．

これを，提案手法では次のように書くことができるようにする．

> 温度計さんは，気温が 10 度になったとき，『寒い！』と言いました．すると，ストーブさんは，電源が入っていなかったので，電源を入れました．

ここで「温度計さん」，および，「ストーブさん」はそれぞれホームネットワークに接続された，部屋内の温度計，および，ストーブである．「温度計さんは... 言いました」「ストーブさんは... 電源を入れました」のように，提案手法ではデバイスをプログラムの中で擬人化して主体的な登場人物として扱う．なお，デバイスの名前には，必要に応じて「太郎」などの固有名詞を割り当てられるようにする．

4.2 日本語をベースとしたデバイスの動作記述

前述のように，本研究では Trigger-Action や ECA ルールなどのルールによりデバイスが制御される環境を前提とする．Trigger-Action ルールについては ECA ルールにおける Condtion 部が省略された物として考えてよい．ECA ルールでは，ある出来事（Event）が起きたとき，ある状態（Condition）であれば，ある動作（Action）を行うというルールを記述する．このような環境においてアプリケーション開発では，各デバイスについて，ECA ルールを記述することになる．提案手法では，これらのルールの連結が「おはなし」となるよう，ユーザには ECA ルールに対応する日本語を記述してもらうようにする．

具体的には，ECA ルールの Event に相当する部分は「〜とき」と表現する．Condition に相当する部分は「〜ので」と表現する．Action に相当する部分は「〜ました」と表現する．

上記の例において，温度計の動作については，

- 気温が 10 度以下になったとき（Event）
- 『寒い！』と言いました（Action）

が，それれぞれ Event と Action に相当する．温度計については，Condition は省略されている．

また，同例におけるストーブの動作は

- すると（Event）
- 電源が入っていなかったので（Condition）
- 電源を入れました（Action）

が，それぞれ，Event，Condition，Action に相当する．なお，ストーブの Event となる「すると」については，記述の冗長性排除のために接続詞を用いる場合であり，詳しくは後述する．

ここで「とき」「ので」「ました」以外の部分は，各デバイスが持つ固有の条件や動作となる．これらの表現について，理想的にはユーザが自由に記述し，自然言語処理技術を用いて自動的にデバイスの具体的な条件や動作（API）に対応づけられることが望ましい．しかし，条件や動作の

© 2017 Information Processing Society of Japan

記述は，同一の事柄に対して多種多様な言い回しが存在する．このため，ユーザの自由な記述を許してしまうと，その内容を判別することは非常に困難な処理となる．

そこで，本研究では，各デバイスのAPIと，それに対応する日本語の表現を，「表現定義ファイル」としてあらかじめデバイスごとに用意するようにする．そして，アプリケーション開発環境の実行時には各デバイスからこの表現定義ファイルを収集して読み込む．

温度計のイベントである「気温が10度になった（とき）」は，ユーザやアプリケーションごとにその設定温度を変更できるようにするため，「10度」という語を含めた完全な定型句ではなく，「気温が○度になった（とき）」という形で「○」の値を自由に変更できるように定義可能とした方がよい．そこで本研究では，それぞれのデバイス機能に対応する日本語表現のなかに変数を定義できるようにする．すなわち，「気温が○度になった（とき）」の「○度」の値は，アプリケーション開発環境上で自由に設定できるようにし，各デバイスのルールに落とし込む際に，具体的な値がパラメータとして定義されるようにする．このため，表現定義ファイルでは，たとえば温度計の「気温が○度になった（とき）」は，温度計が持つAPIの「`OnTemperature(x)`」（xには「○度」の○の値が入る）と対応づけられる．このほか，テレビの「電源が入っていなかったので」は「`isTurnedOn()`」，「電源を入れました」は「`turnOn()`」などのようにデバイスAPIと対応づけられる．

なお，パラメータをともなうデバイスAPIを用いるとき，感覚的な言葉を用いた方がユーザにとっては直感的となる場合がある．たとえば，「気温が10度以下になったとき」は「部屋が寒いとき」と表現した方が，ユーザによってはより分かりやすい表現であることが考えられる．このことへの対応として「寒い」や「暑い」という言葉と，具体的なパラメータ（「10度以下」など）をマッピングすることによってこの解決を図ることができる．また，「寒い」や「暑い」といった感覚はユーザごとに異なるため，ユーザごとにプロファイルを作成する，といった必要も考えられる．このような細やかな対応については，具体的なアプリケーション開発環境側で行えばよい．

4.3 自由な会話の設定

温度計のActionである「『寒い！』と言い（ました）」はメッセージを送るためのデバイスの動作となる．デバイスの動作に関する表現については，デバイスの具体的な動作と結び付ける必要があり，曖昧性を排除するために上記のようあらかじめ決められるものとした．しかし，デバイスの発話（メッセージ）の内容は，イベントとして送信する側と受け取る側で合意が取れればよく，その内容を規定する必要はない．さらに，物語としての広がりを持たせるためには，発話内容はできる限り自由度を持たせた方がよい．

そこで，前述の温度計における温度のパラメータ設定と同様に，デバイスの発話となるメッセージの内容は，ユーザが自由に定義できるようにする．上記の例では，温度計は『寒い！』以外にも，これは『ねぇ，寒くない？』や『冷えてきたね』など，自由な発話ができるようにする．具体的には「○○と言い（ました）」はデバイスの「`sendMessage(”○○”)`」のように対応させ，○○内は自由とする．これによってデバイスどうしの会話内容に対するユーザの自由度を最大限に確保し，様々な物語を作成できるようにする．

4.4 接続詞による冗長記述の排除

デバイスどうしを連携させる場合，あるデバイスのActionとそれをトリガとする他のデバイスのEventを定義する必要がある．前述の例をもとにして厳密に記述しようとすると，

> 温度計さんは，...『寒い！』と言いました (Action)．ストーブさんは，温度計が『寒い！』と言ったので (Event)，...

のように記述して，温度計のActionとストーブのEventを対応づける必要がある．しかし，このような記述は，2回同じ内容の文章が繰り返されるために煩わしく，不自然な文章となってしまう．

このため，本研究では「すると」などの接続詞を活用するようにする．前述の例では，ストーブのEventは「すると」であるとした．文法的に「すると」は順接の接続詞であり，この文章では暗黙的に「温度計の『寒い！』というメッセージを受信したとき」という内容をうける．すなわち，「すると」という接続詞が挿入された場合は，直前のデバイスのActionの結果がそのデバイスのEventとして設定されるように扱う．特に，前述のようにデバイス間の連携はメッセージのやりとりによって行われるため，「すると」が挿入された場合，直前のデバイスによって送信されたメッセージと同一の文字列を直後のデバイスが受信したとき，として設定する．

順接の接続詞（「すると」など）による連携では，メッセージのやりとりを行いながら，デバイスは順番に動作していくことになる．このほか，1つのデバイスの動作に対して，複数のデバイスを動作させたい場合が考えられる．たとえば，温度計が『寒い！』とつぶやいたときに，ストーブとエアコンを同時につける，といった動作が考えられる．このような場合への対応のため，順接の接続師に加えて，並列の接続詞として「同時に」[*1]「また」「ならびに」などの接続詞を用いることができるようにする．たとえば，「温度計さんは... と言いました．すると，ストーブさんは... 電源を入れました．同時に，エアコンさんは... 電源をいれました．」と記述した場合，エアコンのEventはストー

[*1] 「同時に」は厳密には連語であり，接続詞ではない．

© 2017 Information Processing Society of Japan

ブの Event の内容をコピーする．

このほか，あるデバイスの中で 1 つの Event に対して複数の動作を定義する場合も考えられる．このような場合には，「さらに」や「そのうえ」のような累加の接続詞を用いる．たとえば，「エアコンさんは... 電源を入れました．それから，設定温度を 20 度にしました．」などの記述が行えるようにする．ただし，このようなデバイス内における複数の動作定義は，日本語の場合，「て」「し」などの添加や並列の関係を表す接続助詞を用いて表現される場合も多い．そのような表現にも対応できるようにするため，たとえば，「エアコンさんは... 電源を入れて，設定温度を 20 度にしました．」といった記述を可能とする．

4.5 物語への登場人物としてのユーザ

これまで，物語の登場人物として家電などのデバイスを想定し，その動作定義について説明を行った．本研究では，物語における特殊な登場人物して，「ユーザ（人）」を扱えるようにする．これは，メッセージシステム上にユーザからメッセージを送り，デバイスを動作させることを想定したものである．なお，開発時における「ユーザ（人）」には，デバイスと同様，名字や名前などの固有名詞を用いることもできるようにする．

たとえば，上記の例で，温度計ではなく「ユーザが『寒い！』と言った時」というルールを記述したとする．この場合，温度計から『寒い！』というメッセージは送られなくなるが，依然としてストーブは『寒い！』というメッセージを受信したときに電源を ON にする，という動作を記述することができるようになる．すなわち，メッセージシステム上（本研究で用いている実行環境では Twitter）で，ユーザが『寒い！』とメッセージを送れば，ストーブが ON になる，というアプリケーションを作成することができる．あるいは，『おはよう』や『ただいま』『今から帰る』というメッセージに対応し，朝起きた時や家に帰った時の一連のデバイスの動作，家に帰宅する前に家で準備しておいて欲しい内容などを定義するアプリケーションを記述することができる．本研究では，メッセージシステムは Twitter や LINE のような人々がコミュニケーションツールとして用いるものを想定する．「おはなし」をベースとして人と人のコミュニケーションの中にデバイスどうしや，デバイスと人とのコミュニケーションをシームレスに融合させることができる．

5. 作成したアプリケーション開発環境

これまでに述べた手法をホームネットワークアプリケーション開発環境「PlayHouse」として Android OS が稼働するタブレット端末上に実装を行った．PlayHouse の基本画面を**図 1** に示す．画面は絵本の構成を意識し，画面を二等分して絵の部分と文字の部分に画面を分割した．右側はデバイスの動作と連携を文字ベースで定義する部分とした．左側にはアイコンと吹き出しを使ってデバイスどうしの会話内容をチャットのようにグラフィカルに表示する部分を設け，記述されたデバイスどうしの対話の様子をユーザが直観的に理解できるようにした．

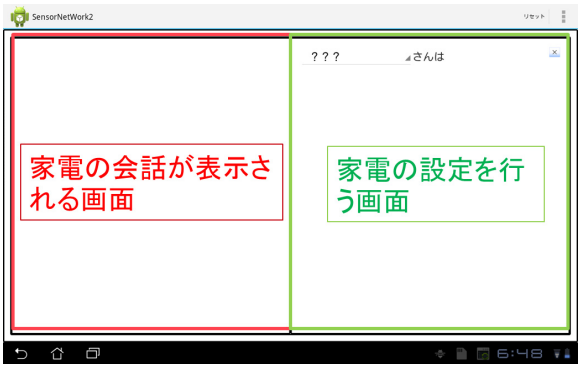

図 1 PlayHouse の初期画面
Fig. 1 Initial screen of PlayHome.

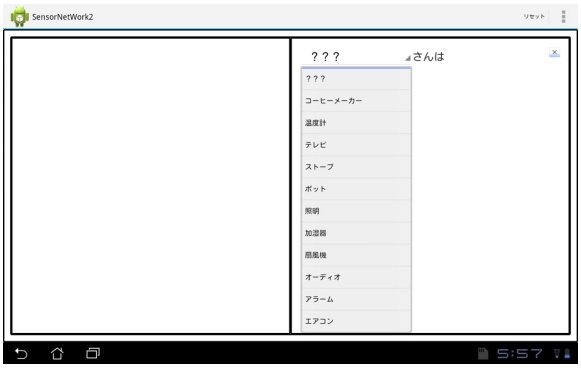

図 2 デバイス選択
Fig. 2 Choosing a device.

5.1 ストーリー例

PlayHouse の動作説明のため，例として温度計，ストーブ，エアコンを使用した例を取り上げる．4.1 節で述べた例に加えて，エアコンを登場させ「気温が 10 度以下になったとき，温度計が『寒い！』と言い，それを聞いたストーブが電源を入れて『すぐ温めますね』と言い，さらにエアコンが電源を入れて『私も協力します！』と言う」というストーリーを作成することにする．

5.2 基本的なデバイスの動作記述

PlayHouse 上で上記のシナリオを実現するにあたり，図 1 の状態から，まずデバイスの選択を行う．右画面にはプルダウンリストボックスが表示されており，このリストボックスにはデバイスの名前が格納されている．プルダウンリストを表示させたときの画面を**図 2** に示す．図 2 では分かりやすさのため，「コーヒーメーカー」や「温度計」など一般名詞が記載されているが，前述のよう各デバイスに固

© 2017 Information Processing Society of Japan

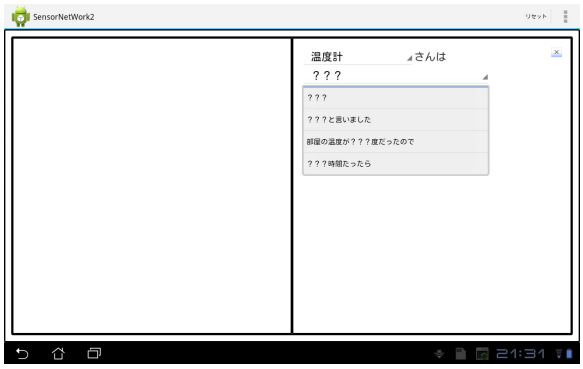

図 3 動作（Event，Condition，Action）の選択
Fig. 3 Choosing an operation (Event, Condition, and Action).

図 4 温度計の Event を選択
Fig. 4 Screen of choosing an event of temperature sensor.

有名詞が割り当てられている場合には，割り当てられた固有名詞が表示される．なお，名前割り当ての方法については本稿では割愛する．

デバイスの選択を行うと，**図 3** に示すように，選択したデバイス（温度計）が保持する Event，Condition，Action の日本語表現が含まれたリストボックスが表示される．このプルダウンから「部屋の温度が？？？度だったので」を選択することで，温度計の Event を設定する．この時の PlayHouse の画面を**図 4** に示す．そして，「？？？」の部分をタップし「10」を設定する．このように，PlayHouse ではデバイスに対して指定可能なイベントや動作をリストボックスでユーザに提示することで，ユーザはデバイスの動作について詳細を把握しておかなかったとしても，動作を指定することができるようにしている．

イベントの選択を行うと，次に，プルダウンメニューとして温度計の Action と Condition からなるプルダウンリストが表示される．ここではストーブとの連携をはかるため，プルダウンリストから「？？？と言います」を選択して温度計に発話させることとする．この選択を行うとエディットボックスが挿入され，「？？？」の部分には自由なメッセージを入れることができる．この例では『寒い！』と入力する．この入力が終わると，**図 5** に示すように，左画面にはデバイスのアイコンが表示され，吹き出しにメッセージの文字が反映される．

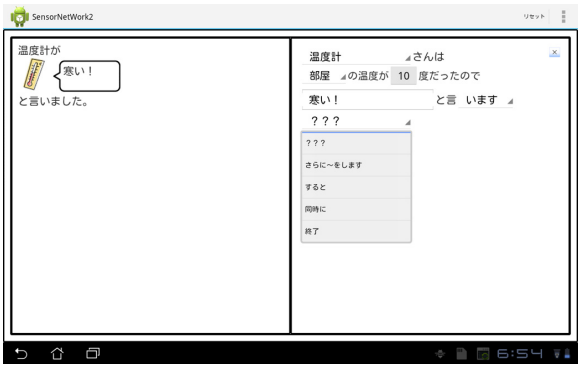

図 5 メッセージの設定画面
Fig. 5 Screen of setting a message.

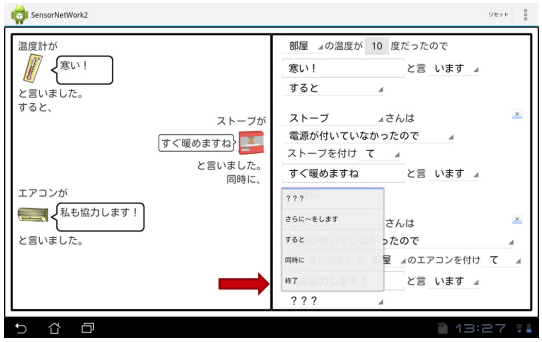

図 6 物語記述（プログラム）を終了する場合
Fig. 6 Finishing programming (story creation).

ここまで完了すると，次に，接続詞が含まれたプルダウンが表示される．接続詞のプルダウンから「すると」を選ぶと，デバイスを選択するプルダウンが挿入される．以下はストーブ，および，エアコンについて同様の手順で設定を進める．最終的に**図 6** のようになり，接続詞のプルダウンにおいて「終了」を選択すると，一連の流れの定義が終了する．

5.3 接続詞の利用

4.4 節で述べたよう，「すると」などの接続詞を用いて Action の記述と Event の記述の重複を避け，文章として自然な流れとなるようにした．また，たとえば，テレビについて「電源を ON にする」という Action と「チャンネルを○○に設定する」という 2 つの Action[*2]を 1 つの Event で行う場合など，1 つのイベントについて 2 つ以上の Action を同時に実行する場合，個々のルールを別々に設定する必要があると，ルールが細分化されすぎてしまい使い勝手を低下させる．このため，PlayHouse 上では，同一のデバイスについて複数の Action の設定を行えるようにした．これは，「さらに」という接続詞による方法と，接

[*2] それぞれ別々の API として提供されることを仮定したが，「電源を ON にしてチャンネルを○○に設定するという」API が提供されても良い．その場合，2 つの Action として分ける必要はない．

© 2017 Information Processing Society of Japan

表 1 PlayHouse における接続詞，接続助詞
Table 1 Conjunction and conjunctive particle on the PlayHouse.

	品詞種類	配置場所	機能
すると	接続詞（順接）	文頭（1 つの文終了後）	直前の文の Action を Event に変換して設定する
同時に	接続詞（並列）*3	文頭（1 つの文終了後）	直前の文の Event をコピーする
さらに	接続詞（累加）	文頭（1 つの文終了後）	直前の文のデバイスに対して Action を追加する
また	接続詞（並列）	Condition 部前（1 つの Condition 部終了後）	文のデバイスに対して Condition を追加する
て（って）	接続助詞（動作の継続）	Action 部の語尾	文のデバイスに対して Action を追加する

続助詞（先に定義された行動の語尾を変化させる）を用いる．図 5 にはリストボックス内に「さらに～をします」という接続詞が表示されており，これを選択した場合と，「さらに,」という接続詞が表示された後，デバイスの Action が含まれたリストが表示されることになる．これにより，「さらに○○をします」という記述で，同時に実行される Action を記載することができる．また，「言います」の「います」の部分もプルダウンリストとなっており，選択すると「います」および「って」の選択肢が現れる．「って」を選択すると，「言『って,』」と次の文を続ける形に語尾を変化させることができる．この場合も同様に，次にデバイスの Action が含まれたリストが表示され，「さらに～をします」を選択した場合と同様，同時に実行する Action が記述できる．

表 1 に PlayHouse 上で実装されている接続詞の種類とその機能を示す．なお，表 1 において，文とは，たとえば「温度計さんは部屋の温度が 10 度だったので『寒い！』と言います」のように，1 つのデバイスに対する ECA ルールの定義となっている文を指す．

表 1 に示した接続詞や接続助詞以外にも，様々な接続詞や接続助詞を考えることができる．しかし，本研究では，順接，並列，累加を表す接続詞や並立，動作の継続/並列を表す接続助詞のみを考える．すなわち，接続詞，接続助詞のうち，流れを崩さずに複数の文や部を追加するもののみを扱う．

本研究で作成対象とするプログラムは，定義されたルールを定義された順に沿って実行していくものを想定する．なお，ECA ルールにおいて，プログラムの分岐は Condition によって行われることになる．たとえば，図 6 に示した「ストーブさんは，電源がついていなかったので，ストーブを付けて，『すぐ温めますね』と言います」という文は，ストーブの電源がついてれば Condition が成立せず，「ストーブを付ける」という Action，および，「『すぐ温めますね』と言う」という Action は行われない．また，このことにともなって以降のルールは実行されなくなる．以降のルールを実行したい場合には，ストーブがついていた場合の動作を記述する必要があるが，これを記述しようとすると，「ストーブさんは，電源がついていなかったので，ストーブを付けて，『すぐ温めますね』と言います．電源がついていない場合は，『もうすぐ温かくなりますよ』といいます」といった，場合分けの記述をしなければならず，1 つの「おはなし」としての自然さを損なうことになる．このため，このような場合分けは，本研究では「別のおはなし」として定義して対応することとする．このように，複雑なフロー制御をともなうルールを記述しようとした場合，複数の「おはなし」を記述する必要がでてしまうが，2 章で述べたよう，既存研究において軽くて単機能なアプリケーションが好まれることが指摘されていることから，本研究では「おはなし」としての文章の自然な流れの作成，および，初心者への分かりやすさを重視した．なお，別の「おはなし」を作成する際，既存のものをコピーするといった作成を容易化するための工夫や，作成されたルールの衝突を検知して通知するといった安全性確保のための工夫が考えられるが，本稿では割愛する．

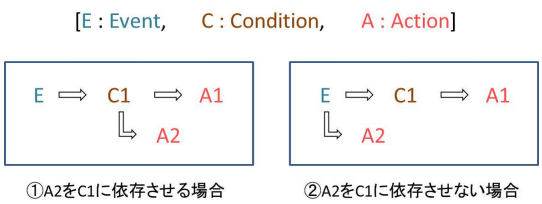

図 7 Condition の有効範囲
Fig. 7 Scope of condition setting.

5.4 Condition のスコープ

接続詞，もしくは，接続助詞の利用において，特に 2 つ以上の Action を設定する場合は，Condition のスコープを考える必要が生じる．たとえば，あるデバイスが Condition1 の状態だったとき Action1 を行う，という設定があったとする．これに続けて，デバイスに Action2 を実行することを設定する場合，Action2 を Condition1 に依存させる場合（**図 7** 左）と，依存させない場合（図 7 右）が考えられる．PlayHouse では，これらをそれぞれ，接続詞による方法と，語尾変化（接続助詞）による方法で切り分ける．Action1，Action2 ともに Condition1 に依存させる場合，語尾変化（接続助詞）による方法を用いる．Action1 のみを依存させ

*3 「同時に」は連語であるが，ここでは接続詞的に用いられるため接続詞としている

© 2017 Information Processing Society of Japan

る場合は，接続詞による方法を用いる．

具体的には，テレビの例において「電源を入れて，チャンネルを○○に設定しました.」と記述すると，このときの Condition（「電源が入っていなかったので」）は「電源を入れる」および「チャンネルを○○に設定する」の両方に影響する．このとき，PlayHouse 上では「電源を入れます」の語尾において，「電源を入れ『て,』」を選択すると，次の選択では，Action のみがプルダウンリストとして表示される．一方，「電源をいれます」として Action を設定した後，接続詞として「さらに～をします」を選択すると，「さらに」という接続詞が挿入される．直前の Condition は，この時点でスコープを失う．次のプルダウンリストでは，Condition と Action の両方が含まれたプルダウンリストが挿入され，新たに Condition を設定することができるようになる．

5.5 削除・修正機能の説明

PlayHouse 上での動作記述の削除は，基本的にはデバイスに対して設定されるひとかたまりのルール（文）ごとに行う．動作記述を行ったデバイスの削除を行うときは，ルール右上に存在する削除ボタン（灰色の×印）を押すことで可能となっている．削除ボタンを押すとユーザに対して確認のメッセージボックスが表示され，その確認にたいして「はい」と答えれば，ルールおよび対応するデバイスの会話部分が削除される．このとき，前後の連携は自動的に削除後の流れの状態にあわせたよう更新される．

PlayHouse 上でデバイスのルール内の細かな修正は，その部分をタップすることによって再度，対応する選択肢が表示され，修正を行うことができる．また，選択肢の中には「？？？」という何も設定されていない選択肢もあり，修正においてこの選択肢が選択されると，現在記述されている内容が削除され，あらたな定義を行うことができる．

6. 評価実験

提案手法の有効性を確認するため，PlayHouse を用いて評価実験を行った．豊橋こども未来館にて開催された「青少年のための科学の祭典」において，小学生以下 18 人を対象に，数分の使い方説明の後，十数分～数十分程度 PlayHouse を使用して自由にアプリケーションの作成を行ってもらった．そして，アンケートを用いて PlayHouse の印象について答えてもらった．実験時の写真を**図 8** に示す．また，後日，研究室内に同様の環境を設置し，著者が所属する大学の学生 8 名（20 歳～24 歳，全員男性．以下，本校では「大学生」と呼ぶ）に同様の実験を行い，アンケートに答えてもらった．大学生は全員，情報・知能工学科の学生であり，情報技術を専門として学ぶ学生たちである．これらの結果の比較を行うことにより，本研究で目的とする，情報技術に不慣れな子供であっても，楽しみながら容易にホームネットワークアプリケーションを作成できる環境の構築ができたかどうかを確認する．

図 8 実験時の写真
Fig. 8 Scene of the exeperiment.

6.1 実験時の実行環境

評価実験において，我々が開発を行っているセンサネットワークアプリケーション実行環境 [20] を RaspberryPi [22] 上で動作させた．3 章で述べたように，この実行環境は本研究で提案する手法の想定環境条件を満たす．

この実行環境において，各デバイスは ECA ルールに基づいて動作する．また，メッセージングシステムには Twitter を用いている．各デバイスはそれぞれ Twitter のアカウントを所持するとともに互いにフォローし合っており，他のデバイスがつぶやいたメッセージを受信することができるようになっている．メッセージシステムに Twitter を用いることによって，通常の（人が用いる）Twitter クライアントでデバイス間のメッセージのやりとりを確認することができる．また，人が呟いたメッセージをデバイスに受信させて，これをイベントする動作定義が行えるようになっている．また，メッセージの履歴がログとして自動的に保存されるため，後からデバイス間のやりとりを振り返ることが容易となっている．これによりアプリケーションの動作を追跡し，デバッグが行いやすい環境になっている．

制御対象となるデバイスとして，空気清浄器，IH クッキングストーブ，オーディオ，扇風機，電気ストーブ，テレビ，コーヒーメーカー，電気ポット，加湿器，照明（電球）を用意した．そして，これらに上記の実行環境を付与して PlayHouse からルールを与えられるようにし，ユーザは作成したルールを各デバイスに反映させて動作確認を行えるようにした．また，目覚まし時計（アラーム），および，温度計は仮想のデバイスとして PC 内で作成し，同様に PlayHouse からルールを定義して動作するようにした．PlayHouse は前章で述べたよう Android Tablet（Sony Xperia Tablet S および ASUS Pad TF303CL）上に実装を行った．

© 2017 Information Processing Society of Japan

6.2 アンケート

実験における事前説明では，PlayHouse について，デバイスの会話を記述して「おはなし」をつくるとそのとおりデバイスが動くということ，および，デバイスやその動作の記述には各プルダウンメニューから選択すればよいこと，発話部分は自由に記述して良いことのみを説明した．その後，自由に触ってよい旨を伝え，PlayHouse を十分から数十分程度触って貰った．このとき，特に作成するアプリケーションなどの指定は行わなかった．そして，被験者がひととおり PlayHouse 上での動作定義が一段落ついたと感じられた点で，アンケートに答えて貰った．

アンケートでは，PlayHouse の使い勝手や物語記述に対する印象を 5 段階のリッカート尺度で評価してもらった．**表 2** に子供向けアンケートの質問肢を示す．Q1 は “楽しさ”，Q2 は “簡単さ” の主観的評価を行うために設けた質問肢である．Q3 はデバイスの擬人化や自然言語での記述が物語記述の模擬に与える影響を評価するために設けた．Q4 はデバイスの擬人化や連携，デバイスの発話（対話）を自由に記述するという要因が “楽しさ” に影響するかどうかを評価するために設けた．Q5 は物語記述のコンセプトおよび操作について “理解のしやすさ” を評価するために設けた，Q6，Q7 はそれぞれ，友人，あるいは，両親（家の中の人）とのコミュニケーションのきっかけとなりうるかどうかを評価するために設けた．Q8 は使用の “継続性” あるいは “発展性” についての評価のために設けた．なお，Q6，Q7 について，本研究では特に子供がホームネットワークアプリケーションを 1 人で完成させる，ということは想定しておらず，友人や，親と相談をしながら，あるいは，親が確認をしながらアプリケーションを作成する，ということを想定している．

このほか，アンケートには難しいと感じた点や，どのようなアプリケーションを開発しようと思ったか，どのようなアプリケーションを開発してみたいか，開発環境に対して思ったことなど，自由記入欄を設けて記入してもらった．なお，「どのようなアプリケーションを開発しようと思ったか」という項目について記述があった場合は，追加の質問として「そのアプリケーションを思ったとおり作ることができたか」という項目についてやはり 5 段階のリッカート尺度で評価を行ってもらった．また，どのようなアプリケーションを開発してみたいか，ということについては，子供の保護者にも記載してもらった．

後日の大学生向けのアンケートは同じ内容の文言を大人にとって分かりやすいよう変更したものを用いた．たとえば，Q5 は「このツールの使い方を人に教えることができますか？」といった記述に変更されている．

6.3 提案システムに対する子供と大学生との評価比較

アンケートの結果を**図 9** に示す．なお，こども未来館における実験において得られた子供のアンケートについて，被験者の年齢分布は，6 歳：2 名，7 歳：3 名，8 歳：3 名，9 歳：3 名，10 歳：2 名，11 歳：4 名，12 歳：1 名であった．

各アンケート項目について，大学生と子供の間で並べ替え Brunner-Munzel 検定を行った．この検定における各アンケート項目の帰無仮説，および，対立仮説は以下のようになる．

帰無仮説 項目における大学生の主観と子供の主観の間に差はない．

対立仮説 項目における大学生の主観と子供の主観の間に差がある．

また，優位水準は 5%とした．検定の結果，いずれの項目においても有意差を確認することはできなかった．なお，それぞれの項目における p 値と効果量を**表 3** に示す．

表 3 から，Q4，Q5，および Q8 は，p 値が比較的低いとともに効果量も比較的大きい．このため，これらの項目については，有意傾向にあると考えられる．

表 2 アンケート項目
Table 2 Questions in questionarie.

Q1	「おはなし」をつくるのはかんたんだと思いましたか？
Q2	「おはなし」をつくるのは楽しかったですか？
Q3	家のモノたちの「おはなし」を作っているような感じがしましたか？
Q4	モノどうしがおはなしすることは，おもしろいと感じましたか？
Q5	ともだちに使い方をおしえてあげられますか？
Q6	ともだちといっしょに使いたいですか？
Q7	お母さんやお父さんといっしょに使いたいですか？
Q8	これをつかって，べつの「おはなし」を作ってみたいですか？

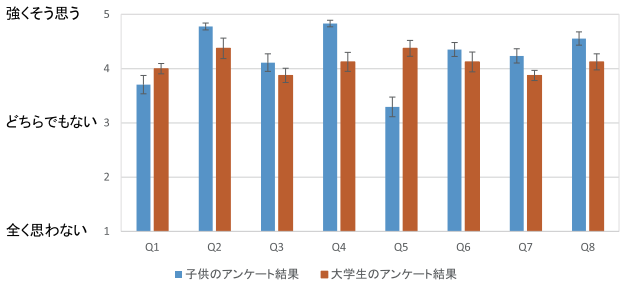

図 9 アンケートの評価結果
Fig. 9 Questionarie result.

表 3 アンケート結果の検定における p 値および効果量
Table 3 P-values and effect sizes of statistical test of qesutionaries.

質問肢	p 値	効果量
Q1	0.707	0.074
Q2	0.137	0.292
Q3	0.288	0.208
Q4	0.060	0.368
Q5	0.073	0.352
Q6	0.656	0.087
Q7	0.340	0.187
Q8	0.076	0.348

© 2017 Information Processing Society of Japan

なお，すべての項目において，子供，大学生とも評点の平均は 3 以上であり，全体的に高い評価を得ている．特に子供に対するアンケートに注目すると，Q2，3，4，6，7，8 では 4 点以上を得ており，特に “Q2：「おはなし」を作るのは楽しかったですか？”，“Q4：モノどおしがおはなしすることはおもしろいと感じましたか？” の 2 つの項目で高い点数を獲得している．一方，“Q1：「おはなし」をつくるのは簡単だとおもいましたか？”，“Q5：友達に使い方を教えてあげられますか？” の 2 つの項目では 3 点台と全体の中では低い点数となった．

以下，各質問肢について考察する．

Q1：「おはなし」を作るのはかんたんだと思いましたか？

子供，大学生とも 4 に近い評価であり，高評価であった．アプリケーション作成過程において，特に被験者が戸惑う様子はみられず，アプリケーションの開発を完遂することができていた．このことから，PlayHouse や提案手法による開発の容易さによって，被験者の思いどおりにアプリケーション開発が行える環境が提供できていると考えられる．なお，大学生に対するアンケートにおいて，Q1 について評価が低かったアンケートには自由記述欄にタブレットの不具合に関する記載があった．そのほかの大学生はほぼ 5 の評価を与えており，実験機材の不具合がなければ評価はより高い物となっていたことが考えられる．また，子供に対するアンケートでは，低学年（6 歳から 10 歳）の子供に低い評価が目立った．そして，自由記述では「文字がなかなかいれられない」「漢字が読めない」という記述がみられた．さらに，実験時においても，被験者が入力に少し戸惑っている様子や親に漢字について聞いている様子がうかがえた．このため，評点を下げている要因は，インタフェースや漢字表示によるものが主であり，本研究の提案自体は有効に機能しているものと考えられる．このことについては，6.4 節でより深く考察する．

Q2：「おはなし」をつくるのは楽しかったですか？

この項目は，子供，大学生ともに非常に高い評価を得ている．この結果から被験者らは情報技術に対する慣れにかかわらず，PlayHouse 上で物語を作成することを楽しめていることが分かり，提案手法によってホームネットワークアプリケーションの開発を楽しみながら行える環境を提供できていることが確認できた．特に子供については，実験時，アプリケーションの作成を終えた後に，作成した物語を読み返してデバイスどうしの会話を見て笑っている子供もいた．さらに，提案するアプリケーション開発環境を自宅でも使いたい，という要望も寄せられた．これらのことから，本手法のホームネットワークアプリケーション開発を「遊び」に落とし込む，ということが有効に働いていると考えられる．

Q3：家のモノたちの「おはなし」を作っているような感じがしましたか？

Q1 同様，子供，大学生とも 4 付近の評点であり，この項目も高い評価を得ている．このことから，本手法で提案するデバイスの擬人化や ECA ルールに対応する自然言語を用いた動作記述が，「物語を記述する」という演出を行うにあたり有効に働くことが確認できた．今回の実験では，デバイスの名前として「テレビ」や「コーヒーメーカー」などの一般名詞を割り当てたため，親近感がわきにくい状態であったと考えられる．固有名詞を用いることでこの評価もさらに改善が可能である，と考えられる．

Q4：モノどうしがおはなしすることは，おもしろいと感じましたか？

子供，大学生とも非常に高い評価となっている．このことから，被験者はデバイスどうしの会話に面白味を感じており，自由な会話記述によるデバイスどうしの連携が有効であることが確認できた．また，Q2 の考察で述べたように，実験中に会話を見て笑っている子もみられたため，本手法で十分にデバイスどうしの連携を楽しめる環境を提供できることが分かった．また，上記のよう本項目には有意傾向がみられ，特に子供に対しては特に会話という演出が有効に働いていると考えることができる．

Q5：ともだちに使い方をおしえてあげられますか？

この項目について，子供では他に比べて低い評価となった．一方で，大学生ではこの項目は 4 を越しており，高い評点を得ている．前述のよう，この項目については，子供と大学生の間で有意傾向がみられている．子供の評点が低い要因として，実験時，被験者が開発環境にふれることができたのは 10 分程度から長くて 30 分程度であり，この時間では「他人に教える」というところまでの習熟は難しかった，ということが考えられる．しかし，実験前の事前の説明では詳しい使い方についてほとんど説明していないにもかかわらず，戸惑うことなく操作を行っている被験者（子供を含む）が多く，提案する開発環境の難易度は高くないと考えてよいと判断できる．大学生についても実際に開発環境にふれることができたのは子供と同程度であり，10 分から 30 分程度であった．しかし，あまり評点が下がらないのは，機材や同種のソフトウェアによる慣れが存在し，全体的な操作の把握に長けているからであると考えられる．

Q6：ともだちといっしょに使いたいですか？

この項目は子供，大学生とも高い評価を得ている．このことから，PlayHouse および提案手法は十分に人に勧められるようなツール（環境）であり，その容易性や楽しさなどを間接的にも示していると考えられる．また，子供については，実験中の観察において，一緒に参加した友達に自分の作った物語を見せあって楽しんでいる被験者も見られた．このことから，提案する開発環境が友達との会話やコ

© 2017 Information Processing Society of Japan

ミュニケーションを活性化するツールとなりうることが確認できた．

Q7：お母さんやお父さんといっしょに使いたいですか？

子供，大学生とも，ほぼ 4 点であるが，大学生は若干低めとなった．子供については，実験中，親と話しながら話の内容を考えていくケースを多く確認することができた．このことから，物語記述のアプローチを用いて，親子が会話しながらホームネットワークのアプリケーションを開発する，といったことを促すことができると考えられる．大学生に対するアンケートでは，この項目は「子供やお父さん，お母さん，友達とのコミュニケーションのきっかけになると思いますか？」という質問の仕方をしており，厳密には子供用アンケートの質問項目とは対応しない．大学生は親元を離れて寮生活もしくは 1 人暮らしをしている人が多く，アンケートに答えている最中，親や子供一緒にアプリケーションを作ることに対して「あまり想像がつかない」といった意見をいう大学生もいた．大学生の評点が若干低めとなったのは，そのような原因が考えられる．

Q8：これをつかって，べつの「おはなし」をつくってみたいと思いましたか？

この項目についても，子供，大学生とも，高い評価を得ている．また，Q2 でも述べたように，PlayHouse を自宅でも使いたい，という要望も寄せられた．このことから，PlayHouse および提案手法は，ホームネットワークのアプリケーション開発を促進できる方法であるといえる．

6.4 提案システムの小学生低学年と小学生高学年との評価比較

Q1 において，子供の評価では，小学校低学年の評価が低めであること，ならびに，その要因は漢字や文字入力などの可能性があることについて述べた．このことについてより詳細な検討を加えるため，子供の被験者を 6 歳から 10 歳（13 名）と 11 歳～12 歳（5 名）の 2 つのグループに分け，これらのグループに対し，Q2 について有意水準 5%で並べ替え Brunner-Munzel 検定を行った．このときの帰無仮説，および，対立仮説は以下のようになる．

帰無仮説 6 歳から 10 歳のグループと 11 歳～12 歳のグループの “簡単さ” の主観評価の間に差はない

対立仮説 6 歳から 10 歳のグループと 11 歳～12 歳のグループの “簡単さ” の主観評価の間に差はある

なお，6 歳から 10 歳のグループの平均は 3.3 点であった．また，11 歳から 12 歳の被験者 5 人の平均点は 4.8 点と非常に高い評点であった．検定の結果，p 値は 0.028 であった．また，効果量は 0.430 であった．すなわち，これら 2 つのグループには有意差があるとみてよい．

これらのことから，PlayHouse は小学校高学年程度には十分に “簡単である” と受け入れられていると考えられる．一方，6～10 歳の子供には現状の PlayHouse は十分に簡単である，とは認識されていないといえる．しかし，前述のようその要因は文字入力や漢字表記に起因するものであると考えられる．たとえば，ひらがなで記述するなどの工夫をすれば，受け入れの幅を広げることができると考えられる．

このほか，大学生には追加のアンケートで「『すると』『同時に』などの（接続詞の）使い方はわかったか」ということについて 5 段階で質問を行った．この結果，平均は 4.5 点であり，非常に高い得点を得た．接続詞の使用についても，混乱なく行えていることを確認した．

以上のことから，提案手法ならびに開発環境は，10 歳以下の子供には文字の入力や，タブレットの操作に難しい部分が残っているものの，情報技術不慣れと考えられる子供達に対しても，十分にホームネットーワークアプリケーションの開発に楽しみを与え，また，簡単に開発を行える環境を提供することができることが確認できた．

6.5 被験者が作成したアプリケーション例

図 10 に，評価実験において被験者（12 歳の女の子）が自身で作成したアプリケーションの画面を示す．以下に，

図 10 被験者（12 歳女の子）が作成したアプリケーションの例

Fig. 10 An example application created by a 12 years old, female subject.

© 2017 Information Processing Society of Japan

作成された内容のテキストを掲載する．

アラームさんは，『おはよう』と言って，7:00 にアラームを鳴らしました．
すると，ストーブさんは，『おはようございます』と言って，部屋をあたため始めました．
すると，テレビさんは，『今日は寒いね』と言って，テレビを付けました．
すると，照明さんは，『ストーブさん，暖めて〜』と言って，明かりを付けました．
すると，コーヒーメーカーさんは，『コーヒー作り開始！』と言って，コーヒーをつくりました．
すると，温度計さんは，部屋の温度が 4 度だったので，『今日は暖かい恰好をしていってね』と言いました．
すると，加湿器さんは，『乾燥しまくってるし』といって，加湿をはじめました．
すると，扇風機さんは，『俺の出番，，，，早く夏になってほしいな』と言いました．
終了

当該被験者は，アンケートにおいて「どのようなお話をつくろうとしたか」という問いに対し，「冬の朝に家で行ってほしいこと」と答え，「おはなしを思った通りに作ることができたか」，という問いに対して，5（強くそう思う）の評価をつけている．また，「Q1. おはなしを作るのは簡単とおもったか」「Q2. おはなしをつくるのは楽しかったか」の問いには，それぞれ 5，4 の評価を回答をしている．また，自由記述にも「楽しくできて，難しくなくてよかった」と回答した．実験時の様子からも，非常に楽しみながら PlayHouse を操作していたことがうかがえ，実験終了時にも「自分の家に持ち帰りたい」と話をしており，非常に気に入った様子がうかがえた．この被験者だけでなく，他の被験者でもルール作成中に笑いながら操作をしている様子が多くみられた．

上記の作成されたルールにおいてデバイスの動作に着目すると，「冬の朝」という状況に対し，適切なデバイスの選択と適切な動作が選択されている．そして，「すると」という接続詞の利用とともに，それぞれのデバイスには適切な発話が定義されており，デバイス間の連携のについても適切に定義されている．1 つのデバイスの動作，および，発話についても，「って」という接続助詞を使って適切に連結されている．特に，最後の扇風機については，発話が定義されているだけであり，扇風機自体の動作は記述されていない．このことは，被験者が扇風機は冬に動作させるべきではないと考えたことを示しており，実際のデバイスの動作と PlayHouse での動作定義の関係をきちんと理解してルールを作成したことを示している．

一方で，実際にデバイスを動作させた際，温度計は 4 度以下に設定されていなかったため，温度計以降のデバイスの動作は生じなかった．このことについて，被験者は最初少し戸惑ったものの，説明により理解をしていた．また，Twitter 上で表示されるメッセージによって，コーヒーメーカーまでの動作で一連のルールの発火が停止していることに気が付くことができていた．実験時にはこのことに対応するルールの変更は行わなかったが，このことから，意図しないデバイスの動作への気づきや，その原因の特定，対応なども容易に行えることが示された．このように，本研究で目的とするアプリケーション開発における「楽しさ」「容易さ」について，物語記述の有効性を確認することができた．

上記以外にも，朝起きたときの一連の動作を作成した被験者が 2 名ほどいた．また，実験の場にはデバイスが揃っていなかったため実現できなかったが，アンケートにおける「つくってみたいアプリケーション」として，「帰宅時に照明の点灯，風呂焚き，夕飯の準備をする」といった一連の動作を自動化するというアイデア（保護者）や，「寝る際に複数の電気製品を OFF にする，あるいは，スリープモードにする」，というアイデア（高校生）もみられた．また，「朝起きる際に，ユーザが起床したかどうかを判断し，最初はステレオで静かな音楽を流し，次にテレビを付け，だんだんと音量を大きくしていく」，といったアイデア（保護者）もみられた．これらは複数（3 つ以上）のデバイスを連携させるアイデアである．「デバイスの連携として，公園にだれか来たらおしえてくれる」（8 歳の男子）や，「子供が学校や塾を出た時間を教えてくれる」（保護者），「帰った時に宿題を教えてくれる」（保護者）といった，一段のデバイス連携でも対応可能となるものも多かったが，上記のよう，本研究における物語記述による手法は，このような一連のデバイス連携に対する発想を与え，また，その実現において特にその利便性や容易性を発揮できるものと考えられる．

実験時，デバイスどうしで口論をする「おはなし」を作成した被験者（9 歳の男の子）もみられた．内容は TV のチャネルについて，ストーブと扇風機がそれぞれ命令してテレビがそれを変更をするという内容であり，実際の動作では，テレビのチャンネルがめまぐるしく変化することとなった．被験者は笑ってそれを見ていたが，被験者はその PlayHouse での動作定義時にそのような動作を考慮してはおらず，PlayHouse 上では「おはなし」を作ることのみに主眼が置かれていたように見受けられた．このように，現状では「おはなし」をすべて作成してからルールを一括してデバイスへ反映させるため，提案手法は，被験者に実際のデバイスの動作とは切り離された PlayHouse の中だけの「おはなし」を作成してしまう場合があることが分かった．しかし，この被験者は「Q1. おはなしをつくるのは簡単と思ったか」「Q2. おはなしを作るのは楽しかったか」という問いに対してそれぞれ 5 の評価を回答するとともに，「Q7.

© 2017 Information Processing Society of Japan

お母さんやお父さんと一緒に使いたいか」という問いに対しても，5という評価を与えている．このため，親と一緒にルールを作る，などの状況により，実際に反映されるルールはより現実的なルールに修正される可能性もある．一方で，「Q6. ともだちと一緒に使いたいか」に対しても5の評価を与えており，このような悪戯ともとれる内容はよりエスカレートする可能性もはらんでいる．今回の実験では見られなかったものの，たとえば冷房とストーブを同時に動作させる，といった矛盾を含むルールを作成してしまう可能性や，場合によっては部屋の温度を非常に高温/低温にしてしまうなど，危険なデバイスの動作を作成してしまう可能性が考えられる．安全性を確保するため，このような矛盾するデバイスの動作や危険な動作など「不適切なルール」をシステム側でチェックし，警告を行ったり，実際に適用するルールから排除する必要性が考えられるが，このことについては今後の課題とする．

7. おわりに

本研究では，家の住人が自身でホームネットワークのアプリケーションで開発を行える環境を提供するため，物語記述によるホームネットワークアプリケーションの開発手法の提案を行った．特に，情報技術にあまり親しみがないと考えられるものの家にいる時間が最も長いと考えられる人々や子供らをターゲットとし，「楽しみながら」かつ「簡単に」アプリケーション開発を行える環境を提供することを目的とした．既存研究によって，多くのホームネットワークアプリケーションは比較的単純な動作で十分であることが示されていることから，本研究で開発対象とするアプリケーションは，Trigger-Action ルールや ECA ルールによる比較的単純なデバイスの動作および連携を対象とした．同様に，アプリケーション開発において「楽しさ」を提供することが重要であることが指摘されており，提案手法では，アプリケーション開発を「おはなしをつくる」という幼少期の「あそび」に帰着させることでユーザに「楽しみ」を与えるようにした．

提案手法では，デバイスを登場人物としてとらえ，デバイスの動作はルールに基づく容易な言葉で定義できるようにした．また，デバイスどうしの連携はデバイスどうしの会話によって表現した．これらを一連の流れとして記述することで，アプリケーションを物語を記述するように開発できるようにした．ECA ルールの Event「～のとき」，Condition「～なので」，Action「～しました」というように表現するようにした．また，自然言語処理の複雑さをさけるため，デバイスの動作は API にすでに対応づけられている日本語表現から選択するようにした．一方，デバイスの会話内容は，制限なく自由に定義できるようにした．また，デバイスの連携では，接続詞を活用することで冗長な表現をさけ，自然な文章の流れでデバイスの動作記述，連携を行えるようにした．

具体的なホームネットワークアプリケーション開発環境として PlayHouse を作成し，PlayHouse を用いて豊橋市こども未来館おけるイベントで評価実験を行い，被験者に自由にアプリケーションを作成してもらってアンケートによって提案手法の有効性を評価した．評価の結果，物語記述によるアプリケーション開発は好意的に受け止められ，楽しんで開発を行えることが分かった．特に，10 歳以下の子供には漢字の判読やタブレット操作について改善が必要なものの，11 歳以上の子供や大人は簡単かつ楽しくアプリケーションを作ることができると感じていることが確認できた．これらのことから，本手法の有効性を確認することができた．

謝辞 本研究は，公益財団法人立石科学技術振興財団の助成を受けて実施されたものである．

参考文献

[1] 阪田史郎：情報家電ネットワークの最新技術動向，電子情報通信学会技術研究報告 MoMuC，モバイルマルチメディア通信，Vol.106, No.359, pp.35–40 (2006).
[2] 丹 康雄：次世代メディアを実現するホームネットワークの現状，電子情報通信学会第 19 回情報伝送と信号処理ワークショップ (2006).
[3] 積水ハウス：Sustainability Report 2011 (2011).
[4] 三菱電機：「大船スマートハウス」でスマートグリッドの実証実験を開始，三菱電機ニュースリリース (2011).
[5] 丹 康雄：ホームネットワークの標準化について，総務省 情報通信審議会 情報通信政策部会 情報通信分野における標準化政策検討委員会 中長期的戦略ワーキンググループ（第 3 回）会議資料 資料 03-05 (2011).
[6] 大和田茂，徳久文彬，市岡陽子，島崎 聡：Kadecot：スマートハウスのいじれる化プラットフォーム，第 20 回インタラクティブシステムとソフトウェアに関するワークショップ（WISS2012）(2012).
[7] 大和田茂：スマートハウスの「いじれる化」を実現するための Kadecot プロジェクト概要，情報処理学会論文誌コンシューマ・デバイス&システム（CDS），Vol.2, No.3, pp.16–22 (2012).
[8] Ur, B., McManus, E., Ho, M.P.Y. and Littman, M.L.: Practical Trigger-Action Programming in the Smart Home, *Proc. SIGCHI Conference on Human Factors in Computing Systems* (*CHI'14*), pp.803–812 (2014).
[9] Widom, J. and Ceri, S.: *Active Database Systems – Triggers and Rules For Advanced Database Processing*, Morgan Kaufmann Publishers (1996).
[10] Terada, T. and Tsukamoto, M.: Smart Object Systems by Event-driven Rules, *Proc. 1st International Workshop on Smart Object Systems* (*SOBS2005*), pp.100–109 (2005).
[11] Owada, S. and Tokuhisa, F.: Kadecot: HTML5-based Visual Novels Development System for Smart Homes, *Proc. 1st IEEE Global Conference on Consumer Electronics* (*GCCE 2012*), pp.17–19 (2012).
[12] Nakamura, M., Matsuo, S. and Matsumoto, S.: Supporting End-User Development of Context-Aware Services in Home Network System, *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing 2012*, pp.159–170 (2013).
[13] Kovatsch, M., Weiss, M. and Guinard, D.: Embedding

© 2017 Information Processing Society of Japan

internet technology for home automation, *Proc. 2010 IEEE Conference on Emerging Technologies and Factory Automation* (*ETFA*), pp.1–8 (2010).
[14] 鈴木孝則：日本語プログラミング言語『和漢』，情報処理学会研究報告計算機アーキテクチャ，Vol.1983, No.44, pp.1–10 (1983).
[15] 岡田 健，松澤芳昭，杉浦 学，大岩 元：教育用プログラミング言語としての「言霊」と「ことだま on Squeak」の試み，情報処理学会教育用プログラミング言語に関するワークショップ 2006 報告集，pp.44–49 (2006).
[16] 日本語プログラミング言語 Mind，入手先 ⟨http://www.scripts-lab.co.jp/mind/whatsmind.html⟩.
[17] 日本語プログラミング言語なでしこ，入手先 ⟨http://nadesi.com/top/⟩.
[18] Resnick, M., Maloney, J., Monroy-Hernndez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B. and Kafai, Y.: Scratch: Programming for All, *Comm. ACM*, Vol.52, No.11, pp.60–67 (2009).
[19] IFTTT: Do more with the services you love, available from ⟨https://ifttt.com/⟩.
[20] 大村 廉，佐々木遼平：Twitter と ECA ルールによるセンサネットワークアプリケーション基盤の構築，マルチメディア，分散，協調とモバイルシンポジウム (DICOMO2012) (2012).
[21] Papamarkos, G., Poulovassilis, A. and Wood, P.T.: Event Condition Action Rule Languages for the Semantic Web, *Workshop on Semantic Web and Databases*, pp.309–327 (2003).
[22] Foundation, R.P.: RaspBerryPi, available from ⟨http://www.raspberrypi.org/.⟩

大村 廉 （正会員）

1999 年慶應義塾大学理工学部電気工学科卒業．2001 年同大学大学院前期博士課程修了．2004 年同大学院後期博士課程修了．博士（工学）同年株式会社国際電気通信基礎技術研究所入社．2007 年慶應義塾大学大学員理工学部情報工学科助教．2010 年豊橋技術科学大学情報・知能工学系講師．現在に至る．センサネットワーク，ウェアラブルコンピューティング，システムソフトウェア等の研究に従事．電子情報通信学会，IEEE，ACM 各会員．

© 2017 Information Processing Society of Japan