

卒業論文 2007年度(平成19年度)

自然言語による知的環境対話システムの構築

指導教員

慶應義塾大学環境情報学部

徳田 英幸

村井 純

楠本 博之

中村 修

高汐 一紀

重近 範行

湧川 隆次

Rodney D. Van Meter III

慶應義塾大学 環境情報学部

伊藤 友隆

tomotaka@ht.sfc.keio.ac.jp

卒業論文要旨 2007 年度 (平成 19 年度)

自然言語による知的環境対話システムの構築

近年、高性能になった小型計算機やセンサなどを家電機器や環境に埋め込み、人々が意識しない形でコンピュータを利用するユビキタスコンピューティングという計算機利用のあり方が盛んに研究されている。

本研究ではこれら計算機やセンサなどを内蔵したモノを知的なモノという意味でスマートオブジェクトと呼称する。また、それらが散りばめられた空間を知的な空間と見なし、スマートスペースと呼称する。現在、高度に情報化された家電機器や携帯情報端末が登場してきており、現在のこの状況はユビキタスコンピューティング環境への過渡期と言える。そんな中我々はこれらのスマートオブジェクトに対して制御であったり、情報取得といった作用をする際、原始的なインタフェースであるボタンやボリュームを使うことが未だに多い。

本研究では、これらスマートスペースであったり、スマートオブジェクトに対しての作用を行う際、自然言語を用いた対話的インタフェースを用いることで、ユーザの操作方法の学習コストや、様々な操作インタフェースの操作方法の選択コストを低減することを目的としている。

本論文では、既存の手法によるスマートスペースおよびスマートオブジェクトに対しての作用について、考察と問題提起を述べ、次いで上述の自然言語による対話的インタフェースによるスマートスペースの操作についての意義を述べた。またさらに自然言語を用いた対話的インタフェースを実現するシステム「Smart Space Conversation」を試作した。

Smart Space Conversation システムは電子メールやインスタントメッセンジャなどのインターネットを利用したコミュニケーション手段を通じて、スマートスペースおよびスマートオブジェクトに対しての働きかけができるシステムである。ユーザは友達と会話するのと同様に、Smart Space Conversation を通じてスマートスペースやスマートオブジェクトと自然言語を用いた対話を行うことができる。

自然言語を用いることで、ユーザは日常生活の中で自身が使っている言葉を用いてスマートスペースに対して作用することができ、複雑な機器の操作に比べて、機器操作の習得がし易い。また、対話的なインタフェースを用いることで、インタラクションの中で不明確な点や、処理に必要な項目を補完してゆくことが可能になり、ユーザの目的を遂行することを助ける。

また、実際に構築した Smart Space Conversation システムのアンケートを用いたユーザビリティ評価を行い、アンケート結果を基に評価を行った。

慶應義塾大学 環境情報学部

伊藤 友隆

Abstract of Bachelor's Thesis

Title

Recently, the research of the ubiquitous computing is active. Ubiquitous computing is consisted of small intelligent sensor, intelligent home electronics. In this research, they are called smart object. And the space they exist in is called smart space. We use the smart objects by using remote-controller or button of itself.

The purpose of this researc is to decrease both learning cost of control manipulations and control device switching cost. This thesis proposes natural language dialog interace for alternative control method.

This thesis describes the problems and considerations of existing control methods first, then describes discussion of natural language dialog interface. And I made a prototype of this interface named Smart Space Conversation. We can access smart space and smart objects through internet by using this system. Users can talk to Smart Space Conversation by using natural language like they talk to their friends.

Using natural language decrease the learning cost of control manipulations and device switching cost. And dialog style allows users to clear up what they are meaning step by step in the dialogs.

The evaluation of the system is described in this thesis. I had a usability test of Smart Space Conversation, and asked questions tested subjects. The evaluation was made from the answers.

Tomotaka Ito

**Faculty of Environmental Information
Keio University**

目次

第 1 章	序論	1
1.1	研究背景	2
1.1.1	ユビキタスコンピューティング環境	2
1.1.2	スマートオブジェクト	2
1.1.3	スマートスペース	3
1.2	問題意識	3
1.2.1	スマートオブジェクトごとに必要な既存インタフェース	4
1.2.2	既存インタフェースの学習コストの大きさ	4
1.2.3	未知な操作方法の獲得しにくさ	5
1.2.4	心理的障壁の大きさ	5
1.3	研究目的	7
1.4	アプローチ	8
1.5	本章のまとめ	9
第 2 章	既存システムによるスマートスペースの操作	10
2.1	情報家電の操作	11
2.1.1	情報家電の定義	11
2.1.2	本体備付のボタンなどによる操作性の悪さ	11
2.1.3	リモートコントローラによる操作の問題点	11
2.1.4	ネットワーク経由による操作の完成度の低さ	12
2.2	センサノードの操作	12
2.2.1	アクセス手段の乏しさ	12
2.2.2	情報取得までのプロセスの複雑さ	13
2.3	本章のまとめ	13
第 3 章	自然言語による対話的インタフェースを用いたスマートスペースの操作	14
3.1	自然言語による対話的インタフェース	15
3.1.1	インタフェースのレイヤ構造	15
3.1.2	自然言語	16

3.1.3	コミュニケーション手段	17
3.1.4	対話におけるインタラクション	17
3.2	スマートスペースの操作	17
3.2.1	空間に対する要求	17
3.2.2	スマートオブジェクトに対する要求	18
3.3	本章のまとめ	18
第 4 章	Smart Space Conversation の設計	20
4.1	設計概要	21
4.2	ハードウェア構成	21
4.2.1	対話サービス提供サーバ	21
4.2.2	コミュニケーション手段	22
4.2.3	スマートオブジェクト	22
4.3	ソフトウェア構成	22
4.3.1	仮想スマートスペース構築モジュール	22
4.3.2	メッセージ監視モジュール	23
4.3.3	命令認識モジュール	24
4.3.4	スマートオブジェクト操作モジュール	24
4.3.5	応答モジュール	25
4.4	本章のまとめ	25
第 5 章	Smart Space Conversation の実装	26
5.1	実装環境	27
5.1.1	開発プラットフォーム	27
5.1.2	スマートスペース	28
5.1.3	スマートオブジェクト	28
5.2	ソフトウェアモジュール	32
5.2.1	仮想スマートスペース構築モジュール	32
5.2.2	コミュニケーションモジュール	33
5.2.3	メッセージ監視モジュール	35
5.2.4	命令認識モジュール	35
5.2.5	スマートオブジェクト操作モジュール	36
5.2.6	応答モジュール	37
5.3	本章のまとめ	38
第 6 章	関連研究	39
6.1	スマートスペースの研究	40
6.1.1	Smart Space Laboratory	40

6.2	スマートスペースにおけるユーザインタフェースの研究	41
6.2.1	Tangible Bits	41
6.2.2	u-Photo	41
6.2.3	実世界 GUI による情報家電プログラミング	42
6.2.4	自然言語を用いて家庭機器操作を行う対話システム	42
6.3	スマートオブジェクトを協調させるための研究	43
6.3.1	JINI	43
6.3.2	cogma	43
6.4	本章のまとめ	44
第7章	評価	45
7.1	評価方針	46
7.2	ユーザビリティ評価実験	46
7.2.1	アンケート実施要領	46
7.2.2	被験者	46
7.2.3	アンケート内容	46
7.2.4	アンケート結果	46
7.3	本章のまとめ	52
第8章	まとめ	53
8.1	本研究の研究背景	53
8.1.1	スマートスペース	53
8.1.2	問題意識	53
8.2	本研究の提案	54
8.2.1	研究目的	54
8.2.2	自然言語による知的環境との対話インタフェース	54
8.3	実装	54
8.3.1	文字入力	54
8.3.2	仮想スマートスペースモデル	55
8.3.3	命令認識	55
8.4	評価	56
8.5	今後の課題	56
8.5.1	認識精度向上	56
8.5.2	フィードバック	57
8.5.3	多言語化	57
8.5.4	音声認識技術との統合	57
8.6	本論文のまとめ	57

目次

1.1	アクチュエータとセンサの違いのイメージ図	3
1.2	既存の情報家電操作モデル	4
1.3	自然言語による対話的インタフェースの操作モデル	4
1.4	リモートコントローラの未知の操作方法獲得手順	6
1.5	自然言語による対話的インタフェースの未知の操作方法獲得手順	7
1.6	スマートスペースに対する操作モデル	8
2.1	小型センサノード μ Part	13
4.1	ハードウェア構成図	21
4.2	ソフトウェア構成図	23
4.3	命令認識モジュールの処理の流れ	24
5.1	SSLab に設置された Canon SX50	29
5.2	正面から見た SHARP RE-M210	30
5.3	SSLab 天井に取り付けられた照明	30
5.4	SSLab 側面に取り付けられた Pioneer PDP-504CMX	31
5.5	カップホルダに取り付けた μ Part	33
5.6	X-Bridge	33
6.1	Smart Space Laboratory	40
6.2	u-Photo の利用図	42
7.1	ユーザビリティ評価実験 被験者アンケート内容	47

表目次

1.1	操作方法習得のための学習内容	5
5.1	Lenovo ThinkPad Z60t の仕様	27
5.2	Canon SX50 の仕様	28
5.3	SHARP RE-M210 の仕様	29
5.4	Pioneer PDP504-CMX の仕様	31
5.5	μ Part の仕様	32
7.1	評価アンケート 問 1 回答	47
7.2	評価アンケート 問 2 回答	48
7.3	評価アンケート 問 3 回答	48
7.4	評価アンケート 問 4 回答	49
7.5	評価アンケート 問 5 回答	49
7.6	評価アンケート 問 6 回答	50
7.7	問 7 回答	50
7.8	評価アンケート 問 8 回答	51
7.9	評価アンケート 問 9 回答	51

ソースコード目次

5.1	CommandGateway(抜粋)	36
5.2	PowerControllable(抜粋)	37

第 1 章

序論

本章では，本研究の研究背景と問題意識について述べる．

1.1 研究背景

本節では本研究の研究背景について述べる。

1.1.1 ユビキタスコンピューティング環境

近年，計算機の高性能化や小型化が著しい．単に計算能力を持つだけでなく，環境の情報を取得するためのセンシング機能や，外部のシステムやノードと通信を行うためのネットワーク到達性を持つ小さな計算機，センサーが登場してきている．これら極小の計算機が，小さいだけでなくやがて安価になり，我々の生活空間の中に散りばめられるようになるというのが「ユビキタスコンピューティング環境」の発想である．

Mark Weiser は The Computer for the 21st century [16] で初めてユビキタスコンピューティング環境の概念を提唱した．

ユビキタスコンピューティング環境ではこれらの計算機やセンサーは人に意識されることなく，様々な情報を通信し合い計算しながら，自律協調し我々の生活をより豊かにするために働く．これらの技術はこれまでのディスプレイの目の前に座って使うといった，人間と計算機の接し方を大きく変える可能性があるとして，情報処理の世界でも多くの研究者が研究テーマとしているところである．本研究ではこのユビキタスコンピューティング環境における人と計算機の橋渡しとなる部分を研究領域としている．

1.1.2 スマートオブジェクト

ユビキタスコンピューティング環境では，前述の通り極小の計算機やセンサによって支えられている．本研究ではネットワーク到達性やセンシング能力を持つこれらのノードを「スマートオブジェクト」と定義する．本研究で想定しているスマートオブジェクトはネットワークを利用して遠隔制御可能なテレビや電子レンジ，プロジェクタ，照明などの情報家電や，環境の情報を取得することのできるセンサなどである．

今後ますます計算機や，センサなどの小型化や高機能化が進み，機器間連携の仕組みが充実することが予想される．また，今までは機械やセンサとは無縁だった日常物にも積極的にそれらが入り入れられることが考えられる．慶應義塾大学の徳田らは u-Texture [13] で知的な素材という意味合いで，計算機やセンサ，ネットワーク機能を持つ素材を提案している．

スマートオブジェクトは実世界において物理的なアクションを起こすことのできる，アクチュエータと，なんらかの環境情報を取得することのできるセンサにその役割を分類することができる．また，両方の機能を持ち合わせているハイブリッド型のスマートオブジェクトも存在する．図 1.1 では，アクチュエータとセンサの性質的な違いを図示している．アクチュエータは環境に対して作用するが，センサは環境に対して情報の取得を行う．

本研究では，これらアクチュエータとセンサに対しての人々の能動的な働きかけを補助すること

アクチュエータ

センサ

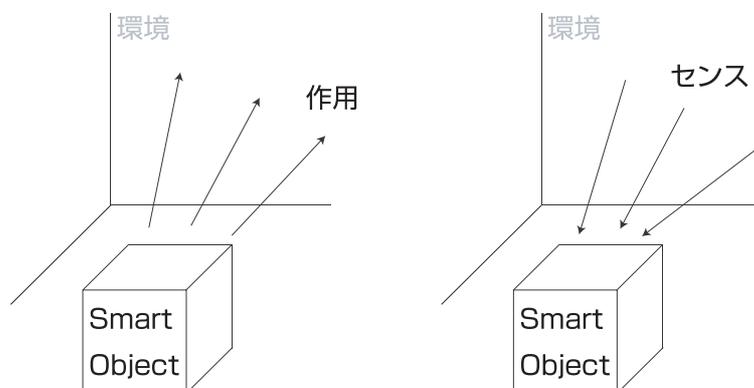


図 1.1 アクチュエータとセンサの違いのイメージ図

を目的としている。

1.1.3 スマートスペース

ユビキタスコンピューティング環境では、多数のスマートオブジェクトが計算能力とネットワークを使って自律協調し、人間の生活をサポートする。ユビキタスコンピューティングが実現されている場所では、必然的に任意の空間には多数のスマートオブジェクトが自律協調しており、その場所にいる人間の作業や生活をサポートしていることになる。スマートオブジェクト単体ではなく、それらが集まって形作る賢さを持つ空間という意味で本研究ではユビキタスコンピューティング環境における任意の空間を「スマートスペース」と定義する。

1.2 問題意識

ユビキタスコンピューティング環境では、人々は多種多様な計算機やセンサーが環境に散りばめられた状態で生活することとなる。そのような環境では多種多様なアクチュエータやセンサに対して働きかける際、共通のインターフェースがないとデバイス全てに対してコントローラや制御ソフト、ボタンなどのインターフェースを利用しなければならなくなり、環境にデバイスが増えれば増えるほどユーザの負荷が増大する。本研究ではこのように多様なデバイスにおいて、共通したインターフェースがないことによるユーザの負荷の増大を問題意識として考えており、本研究では多様なデバイスを単一のインターフェースで扱えることを目標としている。

1.2.1 スマートオブジェクトごとに必要な既存インタフェース

図 1.2 は、既存の情報家電を初めとしたスマートオブジェクトの操作インタフェースでは、家電機器ごとにリモートコントローラを基本とした操作インタフェースが必要とされることを図示している。

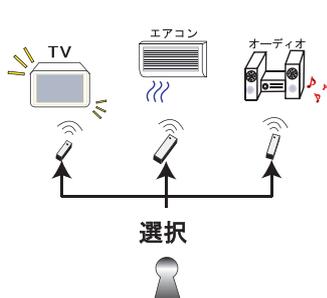


図 1.2 既存の情報家電操作モデル

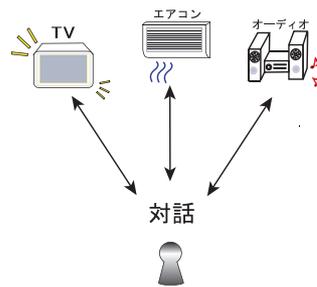


図 1.3 自然言語による対話的インタフェースの操作モデル

図 1.2 の例では、3つのインタフェースを用いて、3つの家電機器を操作する例を示しているが、この例では操作対象の家電機器が切り替わる度に、リモートコントローラを持ち替えなければならない、これは少なからずユーザの負担となる。リモートコントローラを持ち替えるは今手に持っているリモートコントローラを置く、次に使用するリモートコントローラを探す、次に使用するリモートコントローラを手取る、操作を行うといった4つの手順を踏む必要がある。

本研究で提案する自然言語による対話的インタフェースでは、同じ操作環境を継続して利用することができるので、前述のリモートコントローラを置く、探す、手取るという3つの手順を省くことができる。また実際に操作する際のボタンの配置の違いなどの具体的な操作方法を意識せず、同じように自然言語で操作することができる。これを図示したのが図 1.3 である。

1.2.2 既存インタフェースの学習コストの大きさ

表 1.1 では各既存インタフェースと本研究の提案する自然言語による対話的インタフェースの、ユーザが操作方法習得のために覚えなければいけない内容を列挙している。リモートコントローラや本体備付のボタンといった、最も基本的なインタフェースでは、本体の位置や、リモートコントローラの形状といった物理的な情報をまず覚えなければならない。さらにその上でリモートコントローラや本体に備え付けられているボタンの機能を把握しなければならない。これを機器の数だけ覚えなおすことになる。ネットワークによる制御では、物理的な位置を把握しなくても利用可能だが、ソフトウェアの共通化などが未だ十分に図られていないため、インストールや設定などの手間が目立つ。またソフトウェアの使用法も共通化されていないため、リモートコントローラと同じように、使い方を覚える必要がある。

表 1.1 操作方法習得のための学習内容

リモートコントローラ	コントローラの形状, ボタンの配置・機能, 本体の位置
本体備付のボタン	ボタンの配置・機能, 本体の位置
ネットワークによる制御	ソフトウェアのインストール・設定, およびその使用方法
自然言語による対話的インタフェース	コミュニケーション手段, 自然言語

対して, 自然言語による対話的インタフェースでは, 覚えなければならない内容はスマートスペースと対話するためのコミュニケーション手段と, 自然言語の二つのみである. 本研究が提案するのは, スマートスペースとの対話コミュニケーション手段に, 人と人が日常で利用しているコミュニケーション手段を用いることであり, これが達成されることでコミュニケーション手段を覚えるという必要はほぼなくなる. せいぜい必要な作業はアドレス帳にスマートスペース対話のためのアドレスなどを記入する程度で済むと考えられる. 次に自然言語については, 人間が日常に利用する言葉を用いるため, 表現に必要な語彙は元から知っていることになり, 自然言語を覚えなければいけないといったこともない.

これらの点から, 自然言語による対話的インタフェースを用いることによって, 既存のインタフェースに比べて操作方法を習得するまでの学習コストが格段に低くなると言える.

1.2.3 未知な操作方法の獲得しにくさ

既存のインタフェースでは, 未知の操作方法を獲得する際の手順が複雑になりがちである. 図 1.4 では, リモートコントローラをインタフェースとした機器の, 未知の操作方法を獲得する際の手順を UML(Unified Modeling Language) [5] のシーケンス図によって記述したものである. この図を見ると, リモートコントローラをインタフェースとした機器では, 未知の操作方法をユーザが獲得して実行するためには取扱説明書を探し, 取扱説明書の目次を見て目的の項を見つける, 目的の項を読む, リモコンを探す, の計 4 つの手順がある.

一方, 図 1.5 で示された自然言語による対話的インタフェースをインタフェースとした機器操作では, ユーザが未知の操作方法を獲得してから実行するまで, スマートスペースまたはスマートオブジェクトに操作方法を尋ねるといった 1 つの操作手順で済む. また, 両図を比較するとユーザが機器操作を離れて実空間中を探しまわったり, 説明書を読んだり意識の矛先がぶれてしまう. 自然言語による対話的インタフェースでは, そのまま操作を方法を尋ねた上で, 対話を続けなければならないためこのようなことが起こらない.

1.2.4 心理的障壁の大きさ

2007 年現在, 様々な情報家電やネットワークによって連携可能な製品が発売されているが, どれも従来のボタンがついたリモコンといったインタフェースによる操作から脱却できておらず, またそれぞれ製品ごとに異なったリモコンを使う必要がある. とくに最新のテレビやビデオレコー

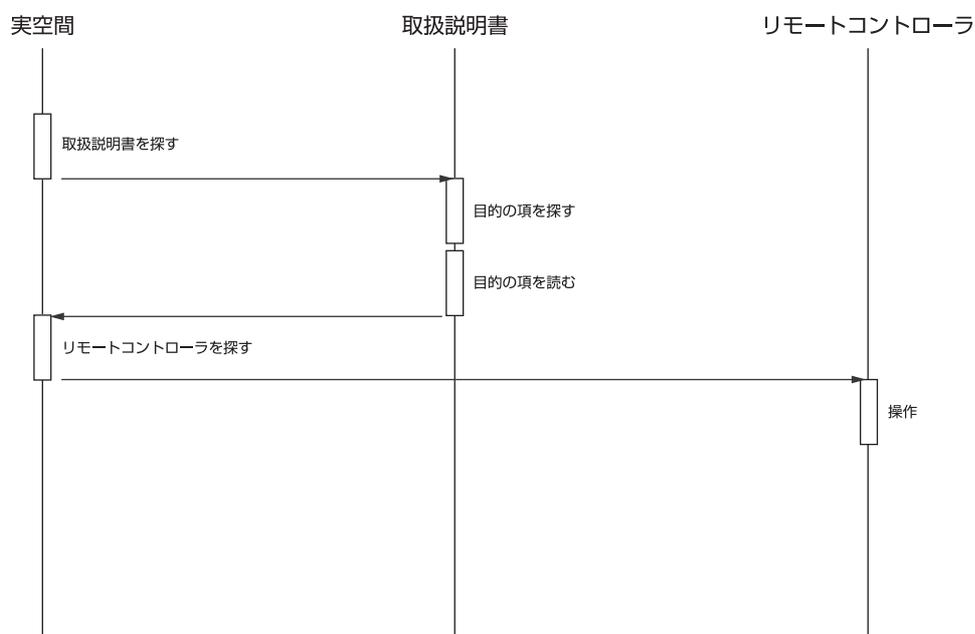


図 1.4 リモートコントローラの未知の操作方法獲得手順

ダーなどは機能の充実に伴い、大量のボタンが搭載されており、分厚い取扱説明を十分読んだ上で正確にボタン操作を行わなければならない、高齢者を初めとした一部のユーザに対して操作に対する心理的障壁を作り出していると考えられる。

とくにユビキタスコンピューティング環境では数多くの機器を連携させ、ユーザが機器の操作そのものを意識しなくて済むことが重要であるため、このような心理的障壁は問題であると考えられる。図 1.6 はスマートスペースに対する、既存のスマートスペース操作モデルと、本研究が提案する自然言語を用いた対話的インタフェースによるスマートスペース操作モデルのイメージ図である。既存のスマートスペース操作モデルでは、ユーザはスマートスペースにあるスマートオブジェクトに対して、それぞれ適切な方法で操作を行う必要がある。対して、本研究が提案する自然言語を用いた対話的インタフェースでは、スマートスペースに対して、まるで人間に話しかけるように対話することでスマートスペース内の各スマートオブジェクトに作用することができる。そのためユーザは機械を操作しているという意識を強く感じずに済む。

自然言語による対話的インタフェース



図 1.5 自然言語による対話的インタフェースの未知の操作方法獲得手順

1.3 研究目的

本研究の目的は、ユビキタスコンピューティング環境におけるスマートスペースに対して、自然言語を用いた対話的なインタフェースによって人々がアクセスすることを可能にすることである。これにより人々は自然言語という使い慣れた表現方法で複雑な手続きなしに、スマートスペースに対して情報の取得を行ったり、スマートオブジェクトに働きかけることが可能となる。

自然言語を使うことでユーザは自分が日常の生活で用いる言語をスマートスペースとの対話に用いることで、機器を操作している意識をなくすことができ、また自然言語ならではの柔軟な表現力がユーザ自身の操作を助ける。

またコミュニケーション手段を問わない対話的インタフェースを用いることで、ユーザがリモートコントローラや専用入力デバイスなどの機器にとらわれることなく、スマートスペースの操作が行える。

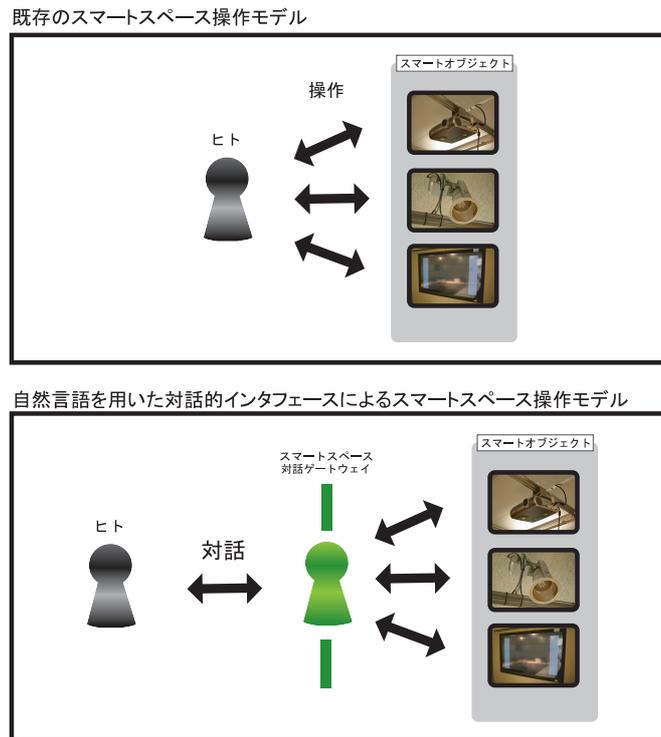


図 1.6 スマートスペースに対する操作モデル

1.4 アプローチ

本論文では、本研究の目的達成のために対話的インタフェースを用いてスマートスペースを操作可能な「Smart Space Conversation」システムを作成する。Smart Space Conversation システムでは人々は電子メールやインスタントメッセージを初めとした、文字ベースの人と対話するためのコミュニケーション手段を用いてスマートスペースへのアクセスが可能となる。

ユーザはこれらのコミュニケーション手段の上で、使い慣れた自然言語による対話をシステムと行うことで、複雑で多様なスマートスペースの操作を小さな心理的障壁で行うことができる。また普段使い慣れている自然言語と、同様に日常生活の中で利用しているコミュニケーション手段を用いてシステムと対話を行うため、本システムを使うための操作方法の学習は、既存のボタンの配列や、入力コマンドに基づいたインタフェースに比べるとその学習コストを大幅に低減することができる。

またネットワークに接続されたシステムを利用することで、リモートコントローラや、本体備え付けのボタンといった操作インタフェースに存在する物理的な制約も存在しないことで、スマートスペースおよびスマートオブジェクト操作のためのインタフェースとしての価値はさらに高まるものと考えられる。

1.5 本章のまとめ

本章では本研究の研究背景，研究の問題意識，研究目的とそれを達成するためのアプローチについて述べた．

研究背景では本研究の研究背景として近年研究が盛んに行われているユビキタスコンピューティングという概念があること，そしてユビキタスコンピューティングが技術の進歩によって実現可能性を近年飛躍的に高めていることを挙げた．

問題意識では，ユビキタスコンピューティング，またはそれに準ずる複数のスマートオブジェクトを擁するスマートスペースという空間におけるコンピューティングにおけるマンマシンインタフェースの古さや学習コストの高さ，物理的制約，心理的障壁などを初めとした使い勝手の悪さを挙げた．

本研究の研究目的は，ユビキタスコンピューティング環境におけるスマートスペースに対して，自然言語を用いた対話的なインタフェースによって人々がアクセスすることを可能にすることである．

そしてそれを達成するためのアプローチとして，文字ベースの日常的に利用可能な，電子メールなどのコミュニケーション手段を利用した対話的なインタフェースをもち，スマートスペース内のスマートオブジェクトに対して作用することのできるシステムを構築することを述べた．

第 2 章

既存システムによるスマートスペース の操作

本章では既存システムによるスマートスペースの操作について述べる。

2.1 情報家電の操作

ここでは既存システムによる，情報家電の操作について述べる．

2.1.1 情報家電の定義

本章における情報家電とは，単純な電子回路だけでなく，内部的に計算機が内蔵されている家電製品と定義する．本研究で作成する Smart Space Conversation システムでは，情報家電はスマートオブジェクトのアクチュエータとセンサという分類において主にアクチュエータに属することになる．従って本節で述べることは，スマートスペースにおけるアクチュエータのアクチュエーションインタフェースの問題点，と言い換えることもできる．

2.1.2 本体備付のボタンなどによる操作性の悪さ

情報家電の古くからある基本的な操作のひとつとして，本体備付のボタンなどによる操作がある．この操作はリモートコントローラにやがて操作方法の主流を奪われたように，多くの問題を抱えている．まず，いちいち操作のたびに情報家電そのものに近づく必要があるという点が最も大きな問題と言える．とくにオーディオプレーヤやテレビ，ビデオレコーダなどではこの問題は重大であり，すぐにリモートコントローラによる操作に主流が切り替わった．

さらに，そもそも本体に大量のボタンを備え付けるのが難しいという問題点があり，これは製品のデザインなどの問題にも関わっている．この問題は現代の高機能化が進んでいる情報家電に対して，備え付けるボタンの数が足りないため本体備付のボタンなどではそもそも機能数に対してボタンの数が不足しており，満足に全ての機能を利用できないという問題につながっている．

2.1.3 リモートコントローラによる操作の問題点

現在の情報家電の最も基本的と思われる操作はリモートコントローラによる操作であると考えられる．現在多く発売されているテレビやビデオレコーダ，オーディオシステムなど人々の生活に密接に関係している一般的な商品はすべてが基本の操作インタフェースとなっている．

本研究がリモートコントローラによる操作において，問題としているのは基本的にはボタンによる操作であることと，距離的制約があること，原則製品ごとに違うリモートコントローラを使わなければならないことである．

ボタンによる操作では，押すボタンがテレビやビデオレコーダに代表される近年の情報家電の高機能化に伴って増加したボタン数により，直感的な操作がしにくいだけでなく，ボタンの押し間違いを防ぐためきちんとユーザ自身が目視による確認をした上でボタンを押して操作を行わなければならない状況が増加している．

またリモートコントローラの多くは赤外線による信号の送信を行っているため，ある程度赤外線

の指向性を考慮した上で情報家電の方へ向けて、操作しなければならない。またあまりにも遠距離で操作した場合は信号が届かなくなるため、距離的制約が少なからず存在する。

操作のためのリモートコントローラは原則製品購入時に付属のものを使う必要があり、複数の情報家電が存在する場合はリモートコントローラも複数あることになり、ユビキタスコンピューティング環境における大量のスマートオブジェクトを想定した場合この操作モデルではスケーラビリティに問題がある。また複数のリモートコントローラの形状などが似ている場合にユーザの選択ミスを誘発しやすい。

2.1.4 ネットワーク経路による操作の完成度の低さ

現在でもネットワーク経路によって制御が可能な情報家電も存在する。わかりやすい例ではネットワークカメラや、ネットワークで操作できるエアコンなどが挙げられる。

しかしながら、これらはネットワーク経路で制御は可能であるものの、専用のソフトウェアを介して行わなければならない、他の情報家電やセンサなどと連動したシステムを構築することは難しい。

ネットワークを介して他の情報家電やセンサなどと連動したシステムを構築するには、お互いがプロトコルと呼ばれる通信方式を理解しあって動作しなければならない。このため、情報家電やセンサ間での通信は、お互いの仕様や、事前の知識に依存しない機器間協調の仕組みが必要である。

2.2 センサノードの操作

ここでは既存システムによる、センサノードに対する操作について述べる。

2.2.1 アクセス手段の乏しさ

まず、既存システムではセンサノードに対してのアクセス手法は非常に限定されており、センサデータを実際に見たり、実際に意味のあるデータとして表示するためにはそのセンサノードに対しての知識が必要不可欠であり、場合によってはセンサデータを取得するためのプログラムをユーザが自分で作成しなければならない場合もある。このように既存システムではセンサノードに対してアクセスの手段が限られていることで、今後センシング能力を持つ情報家電や、エンドユーザが手軽に利用可能なセンサが普及して本格的なユビキタスコンピューティング環境を迎えた際にそのセンシング能力を十分に生かしきれないという問題が考えられる。逆に、センサノードに対して簡便な情報アクセスの手段が確立されていないために、センサノードの一般利用が普及しないとも言えるかもしれない。

本研究ではセンサノードへの情報アクセス手段の乏しさを解決することも目的のひとつである。センサデータを身近に感じることができるようになることで、スマートスペースの価値をよりコンピュータに詳しくない一般のユーザにも認めてもらいたい。

2.2.2 情報取得までのプロセスの複雑さ

センサノードは一般的に、情報を取得するためのインタフェースはプログラムを介したものが一般的であり、エンドユーザが理解しやすい情報を見ることができるソフトウェアはあまり存在しない。ドイツにあるカールスルーエ大学の研究機関、TecO(The Telecooperation Office) [8]によって開発された小型センサノード μ Part(図 2.1) では、同研究機関によって開発された ParticleAnalyzer というソフトウェアを使うことでセンサデータをグラフィカルに表示させることができるが、グラフの波打つ様や、実際にセンサが出力した数字をそのまま見ることができるのみで、人が解釈できる意味の単位に落としこまれてはいないため、センサについての知識が乏しい人がこのソフトウェアを使うことで有益な情報を得られるようになるまでには多少の慣れが少なからず必要である。

このようにセンサデータのアクセスには専用のプログラムを記述するか、または表示ができて非非常にわかりにくいソフトウェアを使うことになることが既存システムの問題点である。

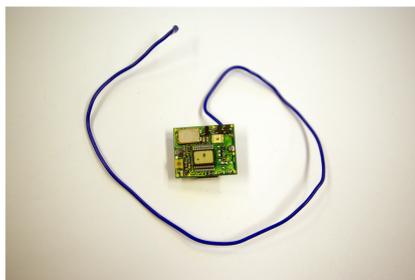


図 2.1 小型センサノード μ Part

2.3 本章のまとめ

本章では既存システムによるスマートスペースの操作について述べた。

既存の情報家電やセンサノードは操作や、それらの内部情報にアクセスする手段に問題があり、よりユーザが直感的に操作できる、新しいインタフェースが必要である。情報家電では既存の本体備付のボタンや、リモートコントローラによる操作が未だに多く、また新しい制御手段としてネットワークを通じた制御などもあるが、一般的でない。センサノードの情報を取り出すには専用のプログラムを作成するか、技術者のように専門的な知識があるものでないと有用に利用できない専用のデータ取得ソフトウェアを利用する以外の方法は事実上ほぼないと言ってよい。

本研究ではこれらの問題を解決すべく、自然言語を用いた対話的インタフェースというアプローチを通じて、ユーザフレンドリなスマートスペースの操作手法を目指している。

第3章

自然言語による対話的インタフェース を用いたスマートスペースの操作

本章では対話的インタフェースを用いたスマートスペースの操作について述べる。

3.1 自然言語による対話的インタフェース

本節では本研究で用いた，自然言語による対話的インタフェースについて述べる．

3.1.1 インタフェースのレイヤ構造

自然言語による対話的インタフェースでは，ユーザが実際に触れる部分から，システムがデータを受け取るところまでインタフェースのレイヤが存在する．ここではこれらインタフェースのレイヤ構造について述べる．

これらのレイヤ構造により，対話というものが各層における機器の違いや，コミュニケーション手段の違いを超えて成り立つことを説明できる．

本研究の目的は自然言語による対話的インタフェースを実現することであり，これらのレイヤ構造に沿った設計を行い，人と人の対話モデルを実現しようとしている．

物理層

自然言語による対話的インタフェースにおける物理層は，ユーザが実際に触れる部分である．例えば，今回本論文執筆にあたって実装したシステムでは，文字入力によるインタフェースであるので，ユーザが実際に触れるインタフェースはキーボードや，電子メールであれば携帯電話などであることも考えられる．

コミュニケーション層

自然言語による対話的インタフェースにおけるコミュニケーション層は，ユーザが利用するコミュニケーション手段である．たとえば携帯電話のよる電子メールの送信という方法でシステムに対して対話を試みる場合，携帯電話の物理的な機種は違うものの，電子メールという共通のコミュニケーションのための規格が統一されている．

文の意味層

自然言語による対話的インタフェースにおける文の意味層は，各コミュニケーション手段から得られた文の，単独での意味である．コンピュータによる自然言語処理を行う場合，得られた入力メッセージからなんらかの文の意味を導き出すことになるが，ここで言及している「意味」とはそのまま，文ひとつそのものの意味であり，文脈を考慮しないものである．

このため，自然言語処理では文脈を考慮した意味というものが存在し，自然言語における対話的インタフェースでは文脈を考慮した意味を考慮した動作をしなければならない．

対話セッション層

文の意味層によって得られた意味も，文脈によって解釈がことなる場合がある，また，文脈がなければそもそも文が意味をなさないこともある．対話セッション層ではそのような会話の前後の流れを保持し，会話全体の意味を正確に取得するために存在する．

会話層

会話層は対話セッション層と文の意味層からなる，システムの内部操作のための命令発効の単位となる会話の意味を作り出す．文脈がないと意味を特定できないものは対話セッション層の助けがあって初めて会話としての意味を持つ．逆に最初から明確に文意が明らかな文章は文の意味層のみでも会話の意味を判断することができる．

3.1.2 自然言語

コマンドラインのような単純でいかにも機械を操作しているといったような対話的インタフェースは古くからあるが，本研究が提案するスマートスペースに対するインタフェースは自然言語による対話的インタフェースである．自然言語とは，人間が日常の中で人との会話のために使う言語であり，日本人が親しんでいる代表的な例をあげるとすれば日本語や英語となる．GUI(Graphical User Interface)の登場まで，機械はCUI(Character User Interface)と呼ばれる文字ベースのコマンドによる対話的インタフェースによって利用されていた．この手法の問題点はコマンドを覚えなければ利用できない点である．GUIはその点，キーボードによるコマンド操作だけではなく，視覚による情報をユーザに提供した上で，コマンド操作だけではなくマウスによる操作をユーザに提供することでコマンドを覚えなければ使えないというCUIの欠点を大きく克服し，専門家以外のユーザが比較的短時間で目的の操作を行うことができるようになった．しかしながら，依然としてキーボードやマウスといった実際にユーザが手を触れる部分のユーザインタフェースは長らく変化していない．結局はこれらのユーザインタフェースの操作に習熟しなければ，コンピュータを使いこなす，自らの目的に役立てることは難しい．

そのため本研究では，これらの伝統的なインタフェースの代わりに，私達が日常的に会話に用いる自然言語をコンピュータ操作のためのインタフェースに用いようと考えた．自然言語をコンピュータが解釈し，自然言語は複雑になりがちで，覚えなければいけない膨大な数のコマンドと比べ，私達が生まれて成長する過程で自然に身につけるものであるため，コンピュータを操作するためだけにまた新しくコマンドを覚える必要がない．これはマウスなどGUIのインタフェースでも同様であり，自然言語による操作が確立されればウィンドウやカーソルなどのGUI特有の概念もわざわざ覚える必要がなくなる．

3.1.3 コミュニケーション手段

自然言語による対話的インタフェースでは、ユーザにとってはコミュニケーションの内容的には対話している相手がスマートスペースであるか、人間であるかを考慮する必要がない。これはユーザが入力する情報も自然言語であり、またシステムの応答も自然言語であるためである。

このため、本研究で提案するシステムとユーザの間のコミュニケーションのための手段は人と人がコミュニケーションのために利用できるものであれば、問題なく利用することができる。本研究のゴールのひとつとして、人と人が対話するように、シームレスに人とスマートスペースが対話できるようになることがある。これを達成するためには人と人が対話するために利用しているコミュニケーション手段のできるだけ多くのものを人とスマートスペースの対話に使えるようにしなければならない。

3.1.4 対話におけるインタラクション

人と人の対話では、一方が喋り、一方が聞くの繰り返しだけではなく、疑問点が湧いた場合話者の話に割り込んで質問をしたり、不明点を感じられた場合もその不明点を不明なままにしないために随時確認を行うなどといった対話の中でのインタラクションが存在する。人とシステムの自然言語による対話を目指す本研究では、この実際の人と人との会話のインタラクションモデルにならない、不明な点を対話の中で段階的に明確にしたり、推測できる部分を相手に対して提示したりというインタラクションを実現するべきであると考える。

3.2 スマートスペースの操作

3.2.1 空間に対する要求

スマートスペースにおける操作では、空間に対する要求というものが存在する。

例えば、空調を制御したり、照明を制御したりという要求が一例である。これらのスマートスペースの空間に対する要求は、単一のスマートオブジェクトだけではなく、複数のスマートオブジェクトが協調しなければ処理することができない。

そのため、ユビキタスコンピューティング環境では単一のスマートオブジェクトだけではなく、複数のスマートオブジェクトを協調して動作させることのできるシステムが求められる。

複数のスマートオブジェクトを協調させて動作させるには二通りのアプローチがあり、一つ目は監督者的な立場のシステムが協調動作させるべきスマートオブジェクトについてそれぞれ操作を行うやり方であり、二つ目はスマートオブジェクト同士が自律的に協調して動作するやり方である。

ユビキタスコンピューティングのあるべき形としては、後者のスマートオブジェクト同士が自律的に協調動作を行うという手法が望まれるが、現在では、各メーカーが各々が作成した仕様に基づいて通信プロトコルなどを作成しているため、難しいと考えられる。

3.2.2 スマートオブジェクトに対する要求

スマートオブジェクトに対するユーザの能動的な要求は、1章で述べたように大きく分けてアクチュエーション（動作要求）と情報取得要求の二つに分けられる。本節ではそれぞれについて順番に説明する。

アクチュエーション

アクチュエーション（動作要求）はスマートオブジェクトに対してなにかの動作を要求することである。たとえばネットワークで制御可能な電子レンジであれば、電子レンジをスタートさせたり、停止させることがアクチュエーションであり、ネットワークで制御できる照明であれば、照明をつけたり消したりすることがアクチュエーションである。スマートオブジェクトに対するアクチュエーションはユーザの要求によって様々であるが、本研究で開発する Smart Space Conversation システムでは、従来のように操作する機器によってリモコンを変えたり、専用のソフトを立ち上げたり、その場に行ってスイッチを押したり、といった行為をしなくて済むようになる。

情報取得要求

情報取得要求はスマートオブジェクトが持つ情報を取得することを要求することである。ここで言う情報とはスマートオブジェクトによって保持されている情報が違うものであり、たとえば電子レンジであれば現在設定されている 600W などの出力の値であったり、センサーであればセンサーデータや自分の ID についての情報である。

ユビキタスコンピューティング環境ではユーザが手軽に、少ない手間ですmartスペースにあるスマートオブジェクトの情報を手に入れられることが求められる。Smart Space Conversation ではユーザは様々なスマートオブジェクトに対して複雑な仕組みを用いずに自然言語で「情報を教えて」とシステムに尋ねるだけでそのスマートオブジェクトの持つ情報を得ることができる。

3.3 本章のまとめ

本章では、自然言語による対話的インタフェースの特徴と、それによるスマートスペースの操作について述べた。

自然言語による対話的インタフェースでは、意味の解釈や、文脈、インタラクションなどの多くの問題を解決する必要がある。また対話そのものをシステムで実現するにあたって、自然言語による対話インタフェースのレイヤ構造を考慮しなければならない。

スマートスペースに対する操作では、ユーザの要求をスマートスペースの空間そのものに対しての要求と、個別のスマートオブジェクトに対する要求に分類することができ、それぞれに対して採るアプローチが変わってくる。スマートスペースの空間そのものに対する要求では、スマートスペースに存在するスマートオブジェクトが協調し合うことで、ユーザの要求を実現しなければなら

ないため、それぞれのスマートオブジェクトに対しての個別の要求よりも実現が難しくなる。

スマートオブジェクトに対する要求は、アクチュエーションと情報取得の二種類に分類され、それぞれ動作の要求と、スマートオブジェクトのセンサデータなどの内部的な情報の取得に相当している。例えば情報家電に対してその役割を果たさせる要求はアクチュエーションであり、センサに対してその観測したセンサデータを取得するための要求は情報取得にあたる。

自然言語による対話的インタフェースを用いたスマートスペースの操作では、これらの機能要件・特徴を考慮した上で構成されるべきである。

第 4 章

Smart Space Conversation の設計

本章では，対話的インタフェースによるスマートスペース操作のためのシステム「Smart Space Conversation」の設計について述べる．以下本章では Smart Space Conversation システムのことを本システムと表記する．

4.1 設計概要

本節では本システムの設計について述べる。

4.2 ハードウェア構成

本システムのハードウェア構成は図 4.1 のようになっている。以下順を追って図中の各要素について説明を述べる。

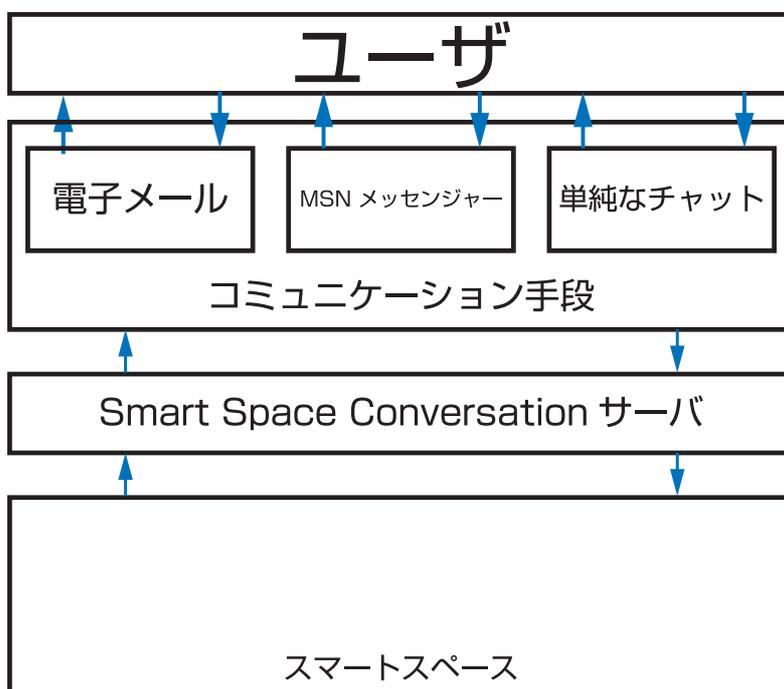


図 4.1 ハードウェア構成図

4.2.1 対話サービス提供サーバ

対話サービス提供サーバでは、ユーザにスマートスペースに対する対話をサービスとして提供する。本システム全体の中でも大きな意味を持っており、ユーザとの対話を行い、ユーザからの対話形式のリクエストを解釈した上で管理している各スマートオブジェクトに対して働きかけるという働きがある。従ってユーザが本システムにおいてスマートスペースに対して情報取得またはアクションを行う際には、必ずこの対話サービス提供サーバを通じて行う必要がある。

対話サービス提供サーバは起動時に後述の仮想スマートスペース構築モジュールによって、自身の記憶領域上に仮想的なスマートスペースのモデルを構築する。これは設定ファイルに基づいたス

スマートオブジェクトの認識であり，起動後に記憶領域上のスマートスペースモデルを変更することは今回の設計では難しくなっている．

仮想スマートスペースモデルを記憶領域に構築した後，メッセージ監視モジュールが設定ファイルによって指定されたコミュニケーション手段を利用し，ユーザからのメッセージ受信の監視を始める．メッセージの受信があると，

4.2.2 コミュニケーション手段

コミュニケーション手段はユーザが実際に触れ，スマートスペースと対話するためのインタフェースとなる部分である．今回実装したバージョンの Smart Space Conversation システムでは以下のコミュニケーション手段をサポートしている．

- 電子メール
- MSN Live Messenger
- 独自プロトコル

今回これらを用いたのは，日常生活におけるそれぞれのコミュニケーション普及の度合いを考慮したためである．電子メールと MSN Live Messenger は特に，インターネットにおける人々のコミュニケーションの標準的な手段として日本ではその地位を得ている．また，単純なチャットによるコミュニケーションはシステムの安定動作を確認するために作成したもっとも基本的な TCP/IP による行指向の対話プロトコルである．これは UNIX を起源とする FreeBSD や，Linux といった OS から telnet という既存のアプリケーションを用いて簡単に利用できる．

4.2.3 スマートオブジェクト

スマートオブジェクトは 1 章で述べたようなセンサーや計算能力やネットワーク到達性を持つ情報家電などを利用した．実際に利用したスマートオブジェクトについては 5 章で述べている．

4.3 ソフトウェア構成

本システムのソフトウェア構成は図 4.2 のようになっている．以下順を追って図中の各要素について説明を述べる．

4.3.1 仮想スマートスペース構築モジュール

ソフトウェア構成図には示されていないが，仮想スマートスペース構築モジュールというソフトウェアモジュールが本システムでは存在する．仮想スマートスペース構築モジュールは，本システムがどのようなスマートオブジェクトがシステムの置かれているスマートスペースにおいて利用可能であるのかを起動時に認識するためのモジュールである．これら利用可能なスマートオブジェク

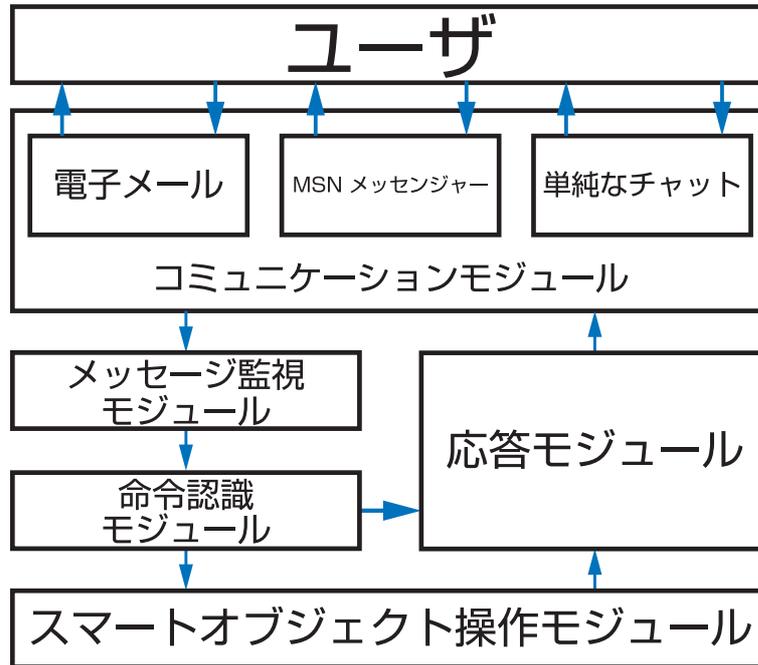


図 4.2 ソフトウェア構成図

トの配置や関係，ネットワーク構成などの情報を仮想スマートスペース構築モジュールでは仮想スマートスペースとして扱う。

仮想スマートスペースは仮想スマートスペース構築モジュールによって起動時に一度だけ作成される。仮想スマートスペースは YAML [10] に則ったスマートスペース設定ファイルを基に対話サービス提供サーバの記憶領域上に構築される。YAML は XML のような，構造化されたデータを表現するための形式のひとつであり，XML より単純な構造がその特徴である。

仮想スマートスペースをシステム上に構築するには，以下の情報が基本的な情報として必要となる。

- メッセージの交換に利用可能なコミュニケーション手段
- スマートスペース内で利用可能なスマートオブジェクト

4.3.2 メッセージ監視モジュール

メッセージ監視モジュールは各コミュニケーション手段を通じてユーザからの対話を監視し，それをシステム上で処理するための受け口となるソフトウェアモジュールである。以下本章ではこのユーザから対話文および，システムからユーザへの対話分をメッセージと表記する。システム起動時に仮想スマートスペース構築モジュール内で監視すべきメッセージングサービスを設定に沿って設定し，監視を開始する。ユーザからメッセージを受信した場合，メッセージ監視モジュールは命

令認識モジュールにユーザから入力されたメッセージを渡し、スマートスペースへの操作として解釈を試みさせる。

4.3.3 命令認識モジュール

命令認識モジュールはメッセージ監視モジュールによって受信されたユーザからのメッセージを解析し、ユーザの要求を解釈し、スマートオブジェクト操作モジュールにスマートオブジェクトの操作を委託する。図 4.3 に命令認識モジュールの処理の流れのイメージを示す。

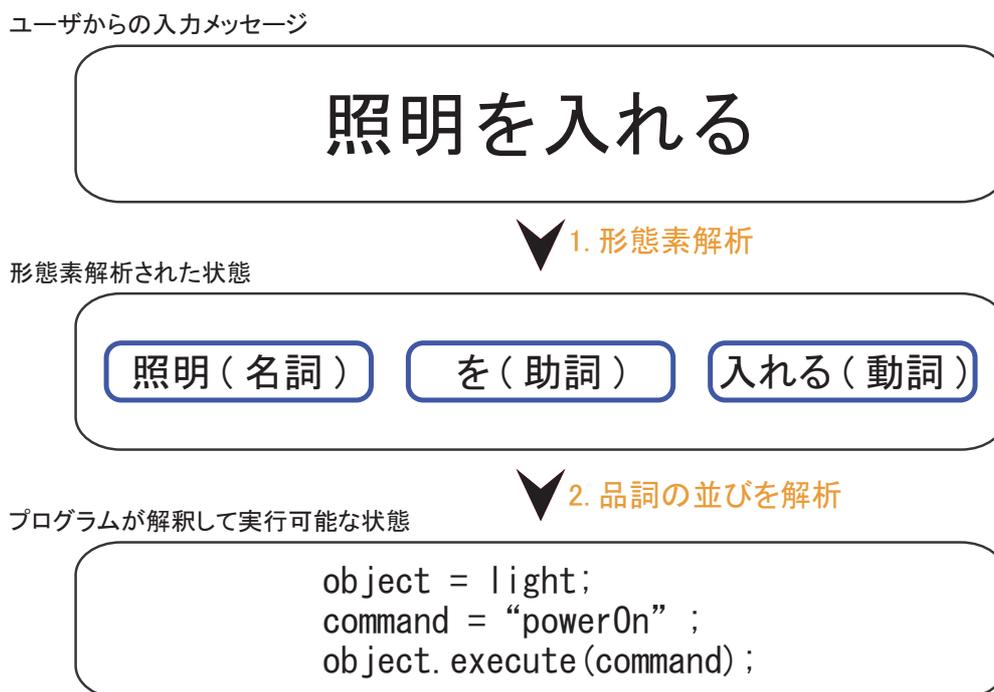


図 4.3 命令認識モジュールの処理の流れ

この命令認識モジュールは本研究における自然言語を用いた対話的インタフェースを実現する重要なソフトウェアモジュールである。

命令認識モジュールはユーザから送信されたメッセージから、「どのスマートオブジェクトに対して」、「何をするのか」を解釈できる必要がある。万が一、ユーザからの入力があいまいであった場合は「どのスマートオブジェクトに対して」、「何をするのか」という点が明確化されないため、応答モジュールを通じてユーザに命令を一意に特定できなかった旨を報告する。

4.3.4 スマートオブジェクト操作モジュール

スマートオブジェクト操作モジュールは命令認識モジュールによって作成されたスマートオブジェクトに対する命令を実行するソフトウェアモジュールである。

スマートオブジェクト操作モジュールが動作する手順としては、まずユーザのメッセージから抽出された「どのスマートオブジェクトに対し」、「何をするのか」という命令を命令認識モジュールから受け取る。次にシステムの起動時に仮想スマートスペース構築モジュールによってシステムの記憶領域上に構築された仮想スマートスペースを参照し、命令を適用するスマートオブジェクトを発見し、命令を実際に発行する。命令によって得られた結果データは、応答モジュールへと渡される。

簡潔にまとめると、命令認識モジュールが作る命令が「命令書」であり、スマートオブジェクト操作モジュールはその命令書に記述されている命令の「実行者」である。

4.3.5 応答モジュール

応答モジュールはスマートオブジェクト操作モジュールが実際にスマートオブジェクトに対して操作や情報取得を行った結果や、ユーザのメッセージから操作を一意に特定不能であった場合はユーザへの聞き直しや提案などの返答を、ユーザが利用しているコミュニケーション手段を通じて返信するためのソフトウェアモジュールである。

4.4 本章のまとめ

本章では、本研究の研究目的を達成するためのシステム、「Smart Space Conversation」の設計について述べた。

ハードウェア構成では、対話サービス提供サーバと、コミュニケーション手段、およびスマートオブジェクトについて述べた。

ソフトウェア構成では、システム実装にあたって設計された各ソフトウェアモジュールの構成と、それぞれの役割について述べた。

第 5 章

Smart Space Conversation の実装

本章では，Smart Space Conversation システムの実装について述べる．以下本章では Smart Space Conversation システムのことを本システムと表記する．

5.1 実装環境

5.1.1 開発プラットフォーム

本節では実装に用いた開発プラットフォームについて述べる．まず開発に用いた環境である，慶應義塾大学徳田・高汐研究室にある実験環境 Smart Space Laboratory(SSLab) [15] について述べ，次に開発に用いたサーバマシンである Lenovo ThinkPad Z60t について述べ，最後に実装に用いた開発言語について述べる．

実験環境

本システムの実装において，知的環境を想定した環境として，慶應義塾大学徳田・高汐研究室にある Smart Space Laboratory を用いた．Smart Space Laboratory では様々なスマートオブジェクトが利用可能であり，本研究の目的を達成するためのシステムの構築と実験，および評価を行う上で非常に適していると判断した．

開発用サーバマシン

本システムでは，開発用のサーバマシンとして Lenovo 社 [4] の ThinkPad Z60t を用いた．本機は標準では WindowsXP が OS としてインストールされているが，本システムの実装に当たって OS はオープンソースとして提供されている Fedora Core [1] を利用した．

開発に利用した ThinkPad Z60t は表 5.1 のような性能を持っている．

表 5.1 Lenovo ThinkPad Z60t の仕様

CPU	Intel Pentium M Processor 2.00 Ghz
メモリ	2048MB
HDD	80GB
オペレーティングシステム	Fedora Core 4

本機の役割は，本システム全体を走らせることであり，具体的には 4 章で述べたようなユーザからのメッセージ受信やユーザへのメッセージ送信，スマートオブジェクトに対する作用や，応答メッセージの決定などを行う．

開発言語

本システムの今回の実装で用いた開発言語は Java 言語である．Java は Sun Microsystems [7] によって開発されたプラットフォームに依存しない，プログラムに可搬性のある言語であり，携帯電話のアプリケーション開発や，Web アプリケーション開発，パーソナルコンピュータ向けのソフトウェア開発まで，現在様々な用途で多くの開発者に利用されている．

本システムの開発においても Java 言語を使用することで、開発環境と実行環境の差を考えなくてよいという大きなメリットを得ることができた。具体的には Windows や Mac OS 上で Java 言語で開発したシステムを Linux の実行環境で動作させることができた。

実行環境に依存しない Java 言語はあらゆるスマートオブジェクトの協調を最終的なゴールとするユビキタスコンピューティング環境の構築という目的に非常によくマッチしており、世界中のユビキタスコンピューティングの研究者の間でも Java によるユビキタスコンピューティングのためのシステムが活発に開発されている。

5.1.2 スマートスペース

本システムの今回の実装に用いたスマートスペースは慶應義塾大学徳田・高汐研究室に Smart Space Laboratory とよばれる次世代スマートスペース研究のための設備である。以下 Smart Space Laboratory のことを SSLab という略称で表記する。

5.1.3 スマートオブジェクト

本節では実装に用いたスマートオブジェクトについてそれぞれ仕様などを記述する。

プロジェクト

今回の実装環境である SSLab で利用可能なスマートオブジェクトのひとつとして、プロジェクトを利用した。

プロジェクトは Canon 製の SX50 というものを利用した。SX50 は表 5.2 のような性能を持っている。

表 5.2 Canon SX50 の仕様

型番	SX50
明るさ	2500(2000)lm
コントラスト比	1000:1
表示可能解像度	1400 × 1050
コンピュータ入力	DVI × 1 / ミニ D-Sub15 ピン
消費電力	標準 290W 静音モード時 240W
外形寸法	幅 284mm × 奥行き 286mm × 高さ 96mm

図 5.1 は SSLab に設置された SX50 の図である。

このプロジェクトにはネットワーク経由でリモート操作したり、画面出力をネットワーク経由で行うという機能はないが、本システムによる操作のためにスギヤマエレクトロン製の学習リモコン Crossam 2+ を利用した遠隔操作システムを Java 言語で構築し、擬似的なスマートオブジェクト



図 5.1 SSLab に設置された Canon SX50

として利用可能とした。これにより、TCP/IP ネットワークの通信によってプロジェクタの電源を遠隔から入れたり、切ったり、映像信号入力ラインを切り替えることが可能になった。

Smart Space Conversation システムでは、「電源を入れる」「電源を切る」「入力を切替える」という命令を認識し、アクチュエーションすることができる。

電子レンジ

今回の実装環境である SSLab で利用可能なスマートオブジェクトのひとつとして、電子レンジを利用した。

電子レンジは SHARP 製の RE-M210 を利用した。RE-M210 は表 5.1.3 のような性能を持っている。

表 5.3 SHARP RE-M210 の仕様

型番	RE-M210
レンジ加熱	900・500・200W 相当
オープン加熱	1380W
グリル加熱	1330W
トースター加熱	1380W
外形寸法	幅 520mm × 奥行き 490mm × 高さ 345mm

図 5.2 は RE-M210 を正面から見た図である。

この電子レンジはシリアル通信による制御用インターフェースを持っており、本システムでの制御を可能とするために開発プラットフォームである Smart Space Conversation サーバにシリアルケーブルで接続し、Java 言語で記述された遠隔操作可能なシステムを構築した。これにより、TCP/IP ネットワークの通信によって電子レンジを遠隔から調理開始したり、調理の停止を行う



図 5.2 正面から見た SHARP RE-M210

ことができるようになった。

Smart Space Conversation システムでは、「調理を行う」「調理を中止する」という命令に対応し、アクチュエーションすることができる。

照明

今回の実装環境である SSLab で利用可能なスマートオブジェクトのひとつとして、照明を利用した。

図 5.3 は SSLab 天井に取り付けられた照明のうちのひとつである。



図 5.3 SSLab 天井に取り付けられた照明

この照明はシリアル通信による制御用インターフェースを持っており、本システムでの制御を可能とするために開発プラットフォームである Smart Space Conversation サーバにシリアルケーブルで接続し、Java 言語で記述された遠隔操作可能なシステムを構築した。これにより、TCP/IP

表 5.4 Pioneer PDP504-CMX の仕様

型番	PDP-504CMX
画素数	水平 1280 × 垂直 768
輝度	1000cd/
外部入力	アナログ RGB , デジタル RGB
外部制御インタフェース	RS-232C D-Sub 9Pin
外形寸法	幅 1218mm × 奥行き 98mm × 高さ 714mm

ネットワークの通信によって照明を遠隔から入れたり、切ったりすることができるようになった。

Smart Space Conversation システムでは、「照明を点ける」「照明を消す」という命令を認識し、アクチュエーションすることができる。

プラズマディスプレイ

今回の実装環境である SSLab で利用可能なスマートオブジェクトのひとつとして、プラズマディスプレイを利用した。

プラズマディスプレイは Pioneer 製の PDP-504CMX というものを利用した。PDP504-CMX は以下のような性能を持っている。

図 5.4 は SSLab 側面に取り付けられた PDP-504CMX の図である。



図 5.4 SSLab 側面に取り付けられた Pioneer PDP-504CMX

このプラズマディスプレイはシリアル通信による制御用インタフェースを持っており、本システムでの制御を可能とするために開発プラットフォームである Smart Space Conversation サーバにシリアルケーブルで接続し、Java 言語で記述された遠隔操作可能なシステムを構築した。これにより、TCP/IP ネットワークの通信によってプラズマディスプレイの電源を入れたり、切ったり、映像信号入力を切替えたりすることができるようになった。

Smart Space Conversation システムでは、「電源を入れる」「電源を切る」「入力を切り替える」という命令を認識し、アクチュエーションすることができる。

μ Part

SSLab 内の様々な箇所に μ Part を取り付け、センサとしてのスマートオブジェクトを操作する機能の実験に用いた。

μ Part の技術的な仕様は表 5.5 のようになっている。

表 5.5 μ Part の仕様

プロセッサ	Microchip 12F675 4Mhz
メモリ	フラッシュメモリ 1.4K バイト, SRAM 64 バイト, EEPROM128 バイト
無線トランスミッタ	19.2kBaud bandwidth, 315Mhz(日本および北米仕様)
無線到達距離	およそ 30m
内臓センサ	運動センサ, 照度センサ (Taos TSL13), 温度センサ (Microchip TC1047)
電源	リチウムコイン型電池 CR1620 型
外形寸法	幅 20mm × 奥行き 17mm × 高さ 7mm
動作温度	-40 から 85 まで

図 5.5 は μ Part を SSLab 内のカップホルダに取り付けた例である。この例では、μ Part の温度センサでカップに入っている飲み物が温かい飲み物か、冷たい飲み物かなどが判断できる。

また、μ Part は無線でセンサデータを飛ばすことしかできず、実際にデータを集めるためには X-Bridge と呼ばれるシンクノードが必要となる。図 5.6 は電源と LAN ケーブルが接続された X-Bridge の例である。X-Bridge は接続されたネットワークに対して、マルチキャストで μ Part のセンサデータを送信することができる。

Smart Space Conversation システムでは、「温度センサ」「照度センサ」「運動センサ」という命令を認識し、センシングした値をコミュニケーション手段を通じてユーザに伝えることができる。

5.2 ソフトウェアモジュール

本節ではソフトウェアモジュールの実装についてそれぞれ詳述する。

5.2.1 仮想スマートスペース構築モジュール

仮想スマートスペース構築モジュールでは、システム起動時に設定ファイルに基づいてシステムの記憶領域上に現在システムが関連付けられているスマートスペース中で有効なスマートオブジェクトの一覧を展開する。また同時に有効なコミュニケーション手段を設定ファイルから判別し、ログインなどの処理を行いスマートスペースに対して関連付ける。この有効なスマートオブジェクトの一覧とコミュニケーション手段とスマートスペースの関連付けのことを Smart Space



図 5.5 カップホルダに取り付けた μ Part



図 5.6 X-Bridge

conversation システムでは仮想スマートスペースと呼称している。

仮想スマートスペースは YAML(YAML Ain't Markup Language) 文法に則った，以下のような形式のファイルによって定義される。

仮想スマートスペース構築モジュールおよび，コミュニケーションモジュールではシステム全体の設定を記述するために YAML 形式 [10] を用いている．これを解析するために Java 言語で利用可能なオープンソースの YAML 解析ライブラリ JYaml [3] を利用した．

5.2.2 コミュニケーションモジュール

コミュニケーションモジュールは，システムとユーザが実際に利用するコミュニケーション手段の橋渡しをするソフトウェアモジュールである．本システムでは実装にあたって，電子メール，MSN Live メッセンジャー，独自プロトコルの三種類のコミュニケーション手段を実装した．

- 電子メール
- MSN Live メッセンジャー
- 独自プロトコル

各コミュニケーション手段をシステムに統合するため，各コミュニケーション手段を取り扱うプログラムは Java 言語の抽象クラス概念を用いて抽象化されたクラス AbstractCommunicator を継承したクラスとして表現され，それぞれ Message というメッセージを抽象化したオブジェクトを送信，受信することができる．

それぞれのコミュニケーションモジュールは，仮想スマートスペース構築の際，メッセージ監視モジュールの一部として組み込まれる．仮想スマートスペース構築モジュールは以下に述べるよう

な形式で設定することで、有効となる。

電子メール

```
- SampleSmartSpace:
  gateways:
    E-Mail:
      receive: pop
      receive-port: 110
      receive-server: pop.example.net
      send: smtp
      send-port: 25
      send-server: smtp.example.net
      smtpauth:
        method: plain
        login: smtpauth-login
        password: smtpauth-password
```

YAML ファイル中のそれぞれの項目について説明を述べる。SampleSmartSpace の項は、定義する仮想スマートスペースの名前を意味している。その中の gateways は、定義される仮想スマートスペースである SampleSmartSpace に対して、関連付けるメッセージ監視のためのコミュニケーション手段である。

電子メールをメッセージ監視のためのコミュニケーション手段として定義するには E-Mail という項を gateways の中に作成する。電子メールの設定として、受信と送信に対する基本的な設定を記述する。

受信に関係する設定は receive, receive-port, receive-server であり、それぞれ受信のためのプロトコル、受信するために接続するサーバのポート、受信するために接続するサーバのドメイン名となっている。

送信に関係する設定は send, send-port, send-server, smtpauth であり、それぞれ送信のためのプロトコル、送信するために接続するサーバのポート、送信するために接続するサーバのドメイン名、SMTP 送信時の認証のための id/pass の組となっている。

MSN Live メッセンジャー

```
- SampleSmartSpace:
  gateways:
    MSN:
      mail:      sample@live.jp
      password:  sample-password
```

YAML ファイル中のそれぞれの項目について説明を述べる。SampleSmartSpace と gateways の項については、電子メールの例と同じである。

MSN Live メッセンジャーをメッセージ監視のためのコミュニケーション手段として定義するには MSN という項を gateways の中に作成する。MSN Live メッセンジャーを有効にするための設定項目は mail, password であり、それぞれ MSN Live メッセンジャーサービスにログインするためのメール、ログインするためのパスワードとなっている。

独自プロトコル

独自プロトコルを有効なコミュニケーション手段として有効にするには、gateways セクションに 'SimpleChat' というセクションを追加する。SimpleChat セクションでは有効なオプションはひとつだけで、システムが TCP/IP 通信による接続を待ち受けるためのポート番号を指定する 'port' のみである。以下に独自プロトコルをコミュニケーション手段を有効な手段として設定した具体的な仮想スマートスペース定義ファイルの記述例を定義する。

- SampleSmartSpace:

```
gateways:
  SimpleChat:
    port: 5555
```

5.2.3 メッセージ監視モジュール

メッセージ監視モジュールは、コミュニケーションモジュールを用いてユーザからシステムへのメッセージ受信を監視するためのソフトウェアモジュールである。メッセージ監視の仕組みとして、本システムでは様々なコミュニケーション手段によるメッセージを実装したコミュニケーションモジュールを使った CommandGateway というクラスのインスタンスをコミュニケーション手段それぞれに対して生成することで行うようになっている。

5.2.4 命令認識モジュール

命令認識モジュールは、メッセージ監視モジュールによって取得したユーザからのメッセージを解釈し、実際に行う処理を決定するソフトウェアモジュールである。自然言語による命令を認識するために、本システムの実装では Java 言語で記述された形態素解析ライブラリ「Sen」 [6] を利用してメッセージを形態素解析した上で、品詞の並びを調べることによって命令を認識するアプローチを採った。命令認識モジュールは、実装上ではスマートオブジェクト操作モジュールと合わせて SmartObject というスマートオブジェクトを表現するクラスを継承したクラスという形で表現されている。これはオブジェクト指向プログラミングの考え方に基いており、「この命令は自分に処理する責任があるのか」といった判断や「処理を行う」という能力はスマートオブジェクト自身

に備わっているものとしてプログラム上表現するべきであると判断したためである。

命令認識の手順としては、仮想スマートスペースが保持する任意の数のスマートオブジェクト全てに対して、形態素解析を行った後の状態のユーザからのメッセージを命令を与えて、処理を行う責任を持っているかどうかを調べる。プログラム中では以下のようなプログラムで処理を行っている。

ソースコード 5.1 CommandGateway(抜粋)

```
1 //実際のユーザからメッセージを受け取る処理を行っている部分
2 public class CommandGateway extends Thread implements CommunicationListener {
3
4     (中略)
5
6     /**
7     * メッセージを受信した際のイベントハンドラ
8     */
9     public void messageReceived(Message message) {
10         //スマートスペースに対して処理を行わせる, その結果を responseとして得る
11         Message response = this.smartSpace.message(message);
12
13         //結果を Messageとして communicatorを通じてユーザに送信する
14         MessageSource sender = message.getMessageSource();
15         response.setMessageSource(message.getMessageSource());
16         response.setMessageDestination(sender.toMessageDestination());
17         this.abstractCommunicator.sendMessage(response);
18     }
19
20     (中略)
21 }
22
23 //スマートスペースによってどのスマートオブジェクトに処理の責任があるかを調べる処理の部分
24 /**
25 * スマートスペースに対するメッセージに対して応える
26 *
27 * @param message スマートスペースに対するメッセージ
28 * @return 応答
29 */
30 public Message message(Message message) {
31     //このスマートスペースが持つすべてのオブジェクトに対して, 処理できるかどうかたずねる
32     Iterator<SmartObject> iterator = this.objects.iterator();
33     while (iterator.hasNext()) {
34         SmartObject obj = iterator.next();
35
36         if (obj.isResponsible(message)) {
37             //処理できるなら処理させる
38             return obj.processCommand(message);
39         }
40     }
41
42     //どのスマートオブジェクトをもってしても, 対応できなかった: 推測し, 提案する
43     return estimateCommands(message);
44 }
```

いずれのスマートオブジェクトをもってしても、対応できないメッセージであった場合は、まず形態素解析されたトークンを調べて、品詞のものだけを選び出し、そのトークンを含むスマートオブジェクトの命令を抽出して、ユーザがやろうとしていることをサポートする。

5.2.5 スマートオブジェクト操作モジュール

スマートオブジェクト操作モジュールはメッセージ監視モジュールによって受信し、命令認識モジュールによって解釈されたスマートスペースへの命令を実際に実行するソフトウェアモジュール

である．スマートオブジェクト操作モジュールも先ほどの命令認識モジュールと同様に，合わせてスマートオブジェクトという概念でプログラム上は表現されている．

スマートオブジェクトに対して「電源操作が可能」や，「チャンネル操作が可能」といった機能的な分類を可能にするために Java 言語の interface という機能を用いた．以下のコードは「電源操作が可能」であるスマートオブジェクトに対して `powerOn`, `powerOff`, `isPowerControlCommand` というメソッドの実装を強制するための `PowerControllable` という interface の例である．

ソースコード 5.2 `PowerControllable`(抜粋)

```
1  /**
2   * 電源制御可能であることを示すインターフェース
3   *
4   * @author tomotaka
5   */
6  public interface PowerControllable {
7      /**
8       * 電源を入れる
9       */
10     public void powerOn();
11
12     /**
13      * 電源を切る
14      */
15     public void powerOff();
16
17     /**
18      * 与えられた命令 Command がインスタンスに対し電源 on/
19      * off を意味する命令である場合真を返す
20      *
21      * @param message
22      * @return
23      */
24     public boolean isPowerControlCommand(Message message);
25 }
```

このような，機能を意味する interface をつくり，実際に `SmartObject` を拡張する実際の処理を行うクラスは，自分が実行可能な処理に応じてこれらの interface を implement することで，それらの機能に対応するという設計になっている．

このため，`SmartSpace` オブジェクトは，自身が保持する `SmartSpace` を列挙し，それぞれに対しユーザが入力したメッセージの処理の責任があるかどうかを確認し，処理の責任がある場合は処理を行わせるという単純な記述で処理を行うことができる．

これによって様々なスマートオブジェクトに対する処理を大量に記述することなく，オブジェクト指向プログラミングの長所を活かした開発が可能となっている．

5.2.6 応答モジュール

応答モジュールは，スマートオブジェクト操作モジュールによって得られた結果や，不明な文意の確認や提案をコミュニケーションモジュールを通じてユーザへ送信するソフトウェアモジュールである．応答モジュールはスマートオブジェクト操作モジュールまたは命令認識モジュールからユーザのメッセージについての実行結果を受け取り，それをもとにコミュニケーションモジュールを通して適切なメッセージをユーザへ返信する．

命令認識モジュールによって分析されたメッセージの文意が不明確で、スマートオブジェクトに対しての命令が一意に定まらない場合、応答モジュールはユーザにメッセージの再入力を促す。

命令認識モジュールによって分析されたメッセージの文意が明確な場合は、命令認識モジュールから命令を受け取ったスマートオブジェクト操作モジュールが、実際にスマートオブジェクトに対する操作を行った結果を応答モジュールに渡す。応答モジュールはその結果を人が理解できる文章の形にしてユーザへ返信する。

5.3 本章のまとめ

本章では Smart Space Conversation システムの実装について述べた。

実装環境の節では、開発プラットフォーム、スマートスペース、スマートオブジェクトについて述べた。

ソフトウェアモジュールの節では 4 章でも述べた、各ソフトウェアモジュールの実装上の設計について述べた。

第 6 章

関連研究

本書では，本研究の関連研究について述べる．

6.1 スマートスペースの研究

本節では、本研究におけるプラットフォームとなる、スマートスペースについての関連研究について述べる。

6.1.1 Smart Space Laboratory

慶應義塾大学の徳田らは”Smart Space Laboratory: Toward the Next Generation Computing Environment” [15] では、次世代のコンピューティングのための空間として、Smart Space Laboratory を提案している。以下、本節では Smart Space Laboratory について、この論文中で使用されている SSLab という表記を用いる。

SSLab ではスマートスペースのコンセプトから、物理的な設計、また実際にスマートスペースのモデルとなる部屋を実際に構築し運用している。

SSLab の物理的な設計における特徴点として、専用のフレームを用いて構築されていること、Box-in-the-Box と呼ばれる二重構造になっていることが挙げられる。専用のフレームを利用することで、センサーを初めとした様々なスマートオブジェクトが設置しやすくなっており、スマートスペースの構築が容易になっている。また Box-in-the-Box と呼ばれる二重構造によって、現在の技術ではサーバなど、バックエンドのシステムを置かないとスマートオブジェクトを動作させたり、センサデータなどを取得したりといったことが難しいという問題にも対処している。内側からは全ての側面は収納として扱われ、必要に応じて機器類を収納の中に収めることができるためである。

図 6.1 は SSLab の実際の様子である。



図 6.1 Smart Space Laboratory

本研究で作成した Smart Space Conversation システムも SSLab を実証実験の環境として利用している。

6.2 スマートスペースにおけるユーザインタフェースの研究

本節では、スマートスペースにおける、ユーザインタフェースについての関連研究について述べる。

6.2.1 Tangible Bits

MIT(マサチューセッツ工科大学) Media Lab. の石井らは論文「Tangible Bits [12]」で、人間の存在する物理的な空間と、コンピュータがデータを取り扱っているデジタルの世界の垣根を越えるための新しいインタフェース・デザインを提案している。

石井らは論文中で現在の GUI によるインタフェースでは我々が生活で培ってきた「周辺感覚 (peripheral sense)」がまったく活かされていないと述べている。これを解消するために Tangible Bits では以下の点に気をつけて研究を行っている、論文中で述べられている。

- 「インタラクティブな表面：机，壁，天井，ドア，窓などの，建築空間の表面を，物理世界とデジタル世界とのアクティブなインタフェースに変換すること。」
- 「ビットとアトムとの結合：手につかみ操作できる物理オブジェクト (graspable objects) とオンライン・デジタル情報をリンクすること。」
- 「アンビエント・メディア (ambient media)：建築空間の中の音，光，影，空気の流れ，水の動きなどの ambient media を，認知の周縁 (periphery) に位置する，サイバースペースとのバックグラウンド・インタフェースとして利用すること。」

これらの要素はユビキタスコンピューティングにおいて、実世界指向インタフェースと呼ばれる概念を実現するための要素である。本研究では、ユーザの操作を極限まで単純化し、最悪の場合システムになにができるか、どうすればいいか自分の言葉で問いかけることができるというインタフェースを目指した。これに対して Tangible Bits ではユーザの触覚や視覚、聴覚などの実際にモノに触れる際の感覚などを活かしたインタフェースの開発が主眼となっている。もちろん Tangible Bits が実現しようとしているインタフェースは、実世界指向インタフェースであるため本研究が問題意識としている点の多くを解消すであろうこともまた事実である。

6.2.2 u-Photo

慶應義塾大学徳田・高汐研究室で研究が行われている u-Photo [14] は、写真を利用したスマートスペース操作のためのインタフェースを提案している。

写真を遠隔制御のための媒体とすることで、FAX や通信などによる可搬性や、言葉で表現した際のスマートスペース内のスマートオブジェクト内の位置指定の曖昧さを取り除くことができるというメリットがある。また、図 6.2 のように制御したいものの写真を撮るだけという操作方法も直感的でわかりやすい。



図 6.2 u-Photo の利用図

しかし、u-Photo の手法では、制御のために専用のマーカをスマートオブジェクトにつけなければいけないこと、常に u-Photo デバイスを使わなければ写真が意味を成さないことなどが問題点として挙げられる。

本研究と u-Photo の類似点として、どちらも様々なスマートオブジェクトを単一のインタフェースで操作することが可能であるという点が挙げられる。

6.2.3 実世界 GUI による情報家電プログラミング

増井俊之は「実世界 GUI による情報家電プログラミング」[18]において、実世界にある紙などを利用して、GUI を模したユーザインタフェースを提案している。

論文中の提案の中では、FieldMouse というデバイスを用いて、バーコードのスクリーンと振るジェスチャを用いて情報家電を操作することが可能であると述べられている。これも u-Photo や Tangible Bits と同じく、従来の手法より直感的であり、既存の操作手法に比べて大幅にユーザの負担を軽減させることができる。

この論文中で増井は「GUI の歴史をふりかえると、時間をかけてボタン/スライダなどといったインタフェースのイデオムが共通知識化されてきたことと、GUI を用いた GUI プログラム開発者環境が整ってきたことが普及の鍵となったようである。」と述べている。本研究や u-Photo、Tangible Bits などを含めて、次世代の計算機利用のためのインタフェースも、ユーザや開発者双方の知識の共通化や認知が進み、開発環境も整うことが普及の鍵となるのは同じであろう。

6.2.4 自然言語を用いて家庭機器操作を行う対話システム

NEC の長田らは、「自然言語を用いて家庭機器操作を行う対話システム」[19]で音声認識を用いたインタラクションによって家庭の機器を制御するシステムのプロトタイプを作成している。本研究に非常に近い研究内容であり、目指すゴールも違いが、本研究では文字による対話を用いた部分、および家電の機器制御だけではなく、広くスマートオブジェクト全体を対象としてとらえ、情報取得もシステムの機能の一つとして含めている点が差異となっている。先行研究として、本研究でも触れたような対話のインタラクションや、命令認識した上での実行モデルなどを参考にした。

6.3 スマートオブジェクトを協調させるための研究

本節では、スマートスペースにおける、スマートオブジェクトを協調させるための関連研究について述べる。

6.3.1 JINI

JINI [2] は Sun Microsystems によって開発された機器間の協調のための技術である。JINI は同 Sun Microsystems 社が開発し、広く一般に使われている Java 言語の上で動作するため、クロスプラットフォーム性がその特徴である。

サービスを提供する機器、本研究のような例であればスマートオブジェクトがその機器にあたるが、それらがネットワークに接続した際に、自分がどのような機能を持っているのかをネットワークにブロードキャストし、自分の存在を示す。またサービス利用側はルックアップサービスと呼ばれる仕組みを利用して、ネットワーク上で利用可能なサービスを発見することができる。

JINI の特徴として、リモートオブジェクトの機能をまったく事前に知ることなく、利用することができることが挙げられる。本研究で作成した Smart Space Conversation システムでは全てのスマートオブジェクトに対し、その存在とその機能を定義しなければならないという問題があった。これは JINI のようなサービス登録、および発見の仕組みを利用することで解消することができ、今後の課題解決のために参考とした。

また JINI と同様の機器間協調プロトコルの研究として Microsoft の Universal Plug and Play (UPnP) [9] がある。UPnP も実際に家庭用ブロードバンドルーターなどを初めとする数多くの製品に組み込まれて利用されており、実績がある。

6.3.2 cogma

名古屋大学の河口らは動的ネットワーク環境における組み込み機器間の連携用ミドルウェアとして、「cogma [17]」を提案している。

河口らは論文中で cogma の設計について、それぞれのスマートオブジェクトに対して以下の機能があることが望ましいとしている。

- 自分自身の基本機能を把握
- 動的に機能の修正・追加が可能
- 外部からの機能制御が可能
- 可能な限り少ないリソースで動作

これらの機能要件は、本研究を進めるにあたっておおいに参考になった。また、システムを実際に作成するにあたって、これらの機能の重要性を再確認することができた。

cogma においては、アドホックネットワークを利用して限りなく少ない手間で各機器を連携させ

ることを実現している．本論文で作成した Smart Space Conversation システムはプロトタイプであるため，スマートスペースやスマートオブジェクトの管理にそこまで時間を割いておらず，簡易的な実装になっており，これらの技術を参考にさらなる利便性を考慮したシステムの改善を行うことが必要であると考えられる．

6.4 本章のまとめ

本章では本研究の関連研究について，スマートスペースの研究，スマートスペースにおけるユーザインタフェースの研究，スマートオブジェクトを協調させるための研究に分類し，それぞれ述べた．それぞれ，本研究に非常に密接に関係している研究領域であり，本研究ではそれぞれの研究領域における先行研究からの知見を活かして，研究およびシステムの開発を行っている．

第7章

評価

本章では，Smart Space Conversation システムの評価について述べる．
本論文を執筆するにあたって，10人のユーザに実際に本システムを使用してもらい，ユーザビリティ評価アンケートに回答してもらった．本章では主にそのアンケート結果を元に，本システムの評価を行う．

7.1 評価方針

Smart Space Conversation システムはユーザインタフェースであるため、評価は主にアンケートによるユーザビリティ評価を主軸にして行った。

また、操作方法習得までの学習コストの低下や、未知の操作方法獲得のしやすさなど、本論文で述べたような特徴を自然言語による対話的インタフェースが本当に持っているかどうか、定量的評価実験によって評価を行った。

7.2 ユーザビリティ評価実験

本節では、評価にあたって実施したユーザビリティ評価実験について述べる。

7.2.1 アンケート実施要領

今回のユーザビリティ評価実験では、被験者に本システムの実装環境である Smart Space Laboratory において、ラップトップマシンを使い、Smart Space Conversation システムを利用してもらった。

被験者は 10 人で、その詳細については被験者の節で詳述する。

被験者にはおよそ 5 分ほど、本システムを使用して SSLab の様々なスマートオブジェクトを操作してもらい、その後アンケートに答えてもらった。アンケートについてはアンケート内容の節で詳述する。

7.2.2 被験者

被験者は 20 代の男性 9 人と、10 代の男性 1 人の合計 10 人である。

後ほどにも述べるが、性別や年代、およびキーボード操作の習熟度に関して偏りが見られる集団であるため、今後さらに多くの、偏りのない被験者に本システムを使用してもらい、評価してもらう必要があると考えられる。

7.2.3 アンケート内容

今回実施したユーザビリティ評価実験では、被験者に図 7.1 のような設問に答えてもらった。

7.2.4 アンケート結果

問 1. 従来のボタンなどによる操作に比べて、自然言語の文字列入力による操作について

問 1. 「従来のボタンなどによる操作に比べて、自然言語の文字列入力による操作について」では、回答の集計は表 7.1 のようになった。

- 問 1. 従来のボタンなどによる操作に比べて，自然言語の文字列入力による操作について
- 問 2. 従来のボタンなどによる操作に比べて，すばやく目的を達成できましたか
- 問 3. 操作方法がわからないときなどに，システムの提案は役に立ちましたか
- 問 4. どんなときにこのシステムを使いたいと思いますか．いくつかでも選択してください
- 問 5. 操作を実行した際のシステムの結果応答はわかりやすかったですか
- 問 6. 意図した命令をきちんと認識しましたか
- 問 7. 操作方法を習得するのに，従来の方法と比べて時間がかかったと思いますか
- 問 8. 日常においてキーボードによるタイピングをする頻度はどの程度ですか
- 問 9. 本システムについて，改善の余地がある点をいくつかでも選択してください
- 問 10. 本システムについて，感想をお聞かせください

図 7.1 ユーザビリティ評価実験 被験者アンケート内容

表 7.1 評価アンケート 問 1 回答

非常に使いやすい	4
使いやすい	6
どちらでもない	0
使いにくい	0
非常に使いにくい	0

問 1. の集計結果では，「非常に使いやすい」と「使いやすい」の合計がこの項目の全てを占めており，「使いにくい」，「非常に使いにくい」という否定的な二項目の回答は選択されなかった．このことから，このシステムを今回試した全ての人が，従来のボタンなどによる操作に比べて使いやすさを感じたことがわかる．とくに，全体の 4 割の被験者は非常に使いやすいと感じており，自然言語による知的環境との対話のユーザビリティは今回の評価実験の結果を見る限り，プロトタイプとして設計・実装された本システムでも十分にそのユーザビリティの高さを発揮できたと考えられる．

問 2. 従来のボタンなどによる操作に比べて，すばやく目的を達成できましたか

問 2. 「問 2. 従来のボタンなどによる操作に比べて，すばやく目的を達成できましたか」では，回答の集計は表 7.2 のようになった．

表 7.2 評価アンケート 問 2 回答

非常に素早く達成できた	2
割と素早く達成できた	6
どちらとも言えない	2
遅くなった	0
非常に遅くなった	0

問 2. の集計結果では、「非常に素早く達成できた」と「割と素早く達成できた」の肯定的な項目の合計が 8 であり、中立的意見の「どちらとも言えない」が 2 だった。「割と素早く達成できた」の 6 票を考えると、被験者全体の過半数以上がこの項目に票を入れていることになり、圧倒的ではないにせよ、目的の達成に要する時間が多少なりとも本システムの利用によって短縮できたという評価が得られた。

問 3. 操作方法がわからないときなどに、システムの提案は役に立ちましたか

問 3. 「問 3. 操作方法がわからないときなどに、システムの提案は役に立ちましたか」では、回答の集計は表 7.3 のようになった。

表 7.3 評価アンケート 問 3 回答

わかりやすく非常に役立った	2
そこそこ役立った	6
どちらとも言えない	2
役に立たなかった	0
わかりにくく非常に役に立たない	0

問 3. の集計結果では、「わかりやすく非常に役立った」と「そこそこ役立った」の肯定的な項目の合計が 8 であり、中立的意見の「どちらとも言えない」が 2 だった。今回使った実装では、入力されたメッセージに応じてその名詞をキーワードに、仮想スマートスペースの内部にあるスマートオブジェクトと照らし合わせ、存在するスマートオブジェクトであれば、関連する命令を表示するというものだった。言い換えれば、目的語ベースで命令の例を取り出すことができた。このアプローチはこの問 3. の集計結果を見る限りでは、2 項目合わせて 8 割の被験者に役に立ったという評価が得られた。

問 4. どんなときにこのシステムを使いたいと思いますか。いくつでも選択してください

問 4. 「問 4. どんなときにこのシステムを使いたいと思いますか。いくつでも選択してください」では、回答の集計は表 7.4 のようになった。

表 7.4 評価アンケート 問 4 回答

部屋の中で操作対象と距離があるとき	7
家の外など遠隔地にいるとき	7
違う部屋にいるとき	4
操作対象の近くににいるとき	1
さまざまな機器を操作しなければいけないとき	7
機器の操作がよくわからないとき	6

問 4. では、ある項目に対する良し悪しではなく、本システムを利用したいと感じる状況を好きな数だけ選択してもらった。

最も票が多かった項目は「部屋の中で操作対象と距離があるとき」、「家の外など遠隔地にいるとき」、および「さまざまな機器を操作しなければいけないとき」がいずれも 7 票で最多であった。「部屋の中で操作対象と距離があるとき」と「部屋の中で操作対象と距離があるとき」は共通して、操作対象と距離があるという点で一致しており、操作対象から距離がある際に、本システムを使うことでネットワークという距離制約のない仕組みの利点が生かされている評価されていると見ることができる。「さまざまな機器を操作しなければいけないとき」の項目では、複数の機器を使う際にそれぞれの操作方法を思い出しつつ、それに沿って機器を操作しなければならないという思考の転換を、本システムの利用によって省くことができると評価されていると考えられる。

問 5. 操作を実行した際のシステムの結果応答はわかりやすかったですか

問 5. 「操作を実行した際のシステムの結果応答はわかりやすかったですか」では、回答の集計は表 7.5 のようになった。

表 7.5 評価アンケート 問 5 回答

非常にわかりやすい	4
わかりやすい	4
どちらとも言えない	2
わかりにくい	0
非常にわかりにくい	0

この設問でも、「非常にわかりやすい」と「わかりやすい」という肯定的な項目で 8 割の票を占め、システムの応答がわかりやすいという評価が得られた。しかしながら、「どちらとも言えない」の項目にも 2 票入っており、まだシステムの結果応答に関して、わかりやすさという点では改善の余地があると思われる。これについては何人かの被験者が感想記述欄にコメントを残してくれているため、後述する。

問 6. 意図した命令をきちんと認識しましたか

問 6. 「意図した命令をきちんと認識しましたか」では，回答の集計は表 7.6 のようになった．

表 7.6 評価アンケート 問 6 回答

非常によく認識した	2
よく認識した	6
どちらとも言えない	1
認識が悪い	1
認識が非常に悪い	0

この設問では，「非常によく認識した」が 2 票，「よく認識した」が 6 票と肯定的な選択肢に多くの票が入っており，命令認識に関しては概ね高い評価を得ていると言えるが，その一方で「認識が悪い」にも 1 票入っており，人によっては今回評価実験に使ったシステムでは，命令認識に関する部分の完成度が不十分だという評価を下していることがわかる．

問 7. 操作方法を習得するのに，従来の方と比べて時間がかかったと思いますか

問 7. 「操作方法を習得するのに，従来の方と比べて時間がかかったと思いますか」では，回答の集計は表 7.7 のようになった．

表 7.7 問 7 回答

従来の方と比べて非常に短時間で操作を習得できた	4
比較的短時間で操作を習得できた	6
どちらでもない	0
比較的時間をかけて操作を習得しなければならなかった	0
非常に時間をかけて操作を習得しなければならなかった	0

この操作方法を習得するのにかかった時間に関する設問では，「従来の方と比べて非常に短時間で操作を習得できた」と「比較的短時間で操作を習得できた」の 2 項目で被験者の全部をカバーしており，被験者全員が従来の方と比べ，比較的短時間で操作を習得できたと評価している．

問 8. 日常においてキーボードによるタイピングをする頻度はどの程度ですか

問 8. 「日常においてキーボードによるタイピングをする頻度はどの程度ですか」では，回答の集計は表 7.8 のようになった．

この設問では，一人を除く全ての被験者が「毎日非常によく使う」を選択している．これは評価実験を行った環境が一般的な場所ではなく，コンピュータを使った授業などがほぼ全てを占めてい

表 7.8 評価アンケート 問 8 回答

毎日非常に良く使う	9
使わない日もあるが、よく使う	1
平均程度	0
あまり使わない	0
まったく使わない	0

る慶應義塾大学湘南藤沢キャンパスであるということが原因のひとつとして考えられる。今後の課題として、キーボード操作の習熟度に偏りが無い被験者に本システムを使用してもらい、評価してもらう必要があるだろう。

問 9. 本システムについて、改善の余地がある点をいくつかでも選択してください

問 9. 「本システムについて、改善の余地がある点をいくつかでも選択してください」では、回答の集計は表 7.9 のようになった。

表 7.9 評価アンケート 問 9 回答

認識精度	8
入力インタフェースの選択肢増加	4
対応機器の増加	7
多言語化	5

この設問では、全ての項目について今回システムを試してもらった被験者の半分以上が改善の余地があるとしている。と「認識精度」は得票が多く、被験者が改善点と評価していると見ることができる。次点で多いものは「対応機器の増加」である。今回実施した評価実験では非常に限られたモデルでの実験であったため、対応機器が少ないと感じたユーザが少なくなかったようだ。

問 10. 本システムについて、感想をお聞かせください

問 10. 「本システムについて、感想をお聞かせください」では、被験者に試してもらった本システムの感想を自由に記述してもらった。以下、目立ったコメントをリスト形式で取り上げ、それについての本研究の今後の予定や、考え方などを併記する。

「複雑な操作の際にはどうするか」 自然言語による複雑な操作では、入力文字列の長さに比例して、急激に意味の解釈の難しさが増大する。そのため、複雑な操作を行う必要がある際は、小さな文の並びに分割し、操作を行うのが現実的であると考えられる。

「言葉のレパートリーを増やして欲しい」 今回のような、ルールファイルに基づいた命令の認識手法では、言葉のレパートリーはルールファイルに追記することで十分対応可能である。

「フィードバックがわかりにくい」 今回実験に用いた実装では非常に単純な提案システムのため、大量の例文が出力され、被験者の混乱を招いたケースがあったようである。より精度を高め、絞り込んで出力する情報を少なくするなどの選択肢が考えられる。

「同じような名詞を何度も打つのが大変。ニックネームなどをつけれたら」 たしかに、入力における冗長性が現在の実際ではあることを認識している。今後の課題として、現在何について作業を行っているのか、という文脈を取得できるようにするといった方面での機能拡張が考えられる。

7.3 本章のまとめ

本章では、本研究の評価として、Smart Space Conversation システムのユーザビリティ評価実験を行い、そのアンケート結果を考察し、評価を行った。評価実験は本システムの実装環境である、SSLab で行い、実際に 5 分ほど被験者自身にインスタントメッセンジャ、単純なチャットを用いたシステムの利用を行ってもらった。被験者の人数は 10 人である。

アンケート結果によると、従来の方法に比べて操作方法を習得するのにかかる時間がすくなく、また使い勝手がよいという評価を得た。そのほか、目的を達成するまでにかかった時間や、意図した命令をきちんと認識したかどうかという認識精度についても、概ね良好な評価を得た。また今後の課題として、認識精度や対応機器の増加という点が、実際に利用してみたユーザからの改善点として挙げられていた。

第 8 章

まとめ

本論文では，Smart Space Conversation システムの設計と実装を行った．本章では本論文のまとめを行う．

まず，本研究の研究背景として，スマートスペースの普及と，それに伴う問題意識についてまとめる．次に，本研究の提案として，研究目的を述べたあと，自然言語による知的環境との対話インタフェースについて述べ，その特徴についてまとめる．次に，本論文執筆に当たって構築した Smart Space Conversation システムの実装面についてまとめる．次に，Smart Space Conversation システムの評価実験を通して得られた評価についてまとめ，最後に今後の課題についてまとめる．

8.1 本研究の研究背景

8.1.1 スマートスペース

本研究の研究背景として，スマートスペースの普及が挙げられる．知的環境とは，小型のセンサーノードや，情報処理能力やネットワーク到達性を持った情報家電が存在する空間のことである．近年の目覚ましい情報技術およびネットワーク技術の進歩により，これらのスマートオブジェクトをネットワーク越しに操作したり，情報を取得したりという状況が生まれつつあり，また今後そういった機会はますます増加するものと考えられる．

8.1.2 問題意識

スマートスペースにおいて，様々なスマートオブジェクトを操作する際，現在ではまだボタンを伴うリモートコントローラによる操作が一般的である．リモートコントローラには，以下のような問題点がある．

- 位置的制約: 操作位置から操作対象が目視で確認できること
- 距離的制約: 赤外線が届く距離であること
- 機器的制約: 機器はそれぞれ専用のコントローラのみでしか制御できない
- 学習コスト: それぞれの機器に対して、それぞれの操作方法を覚える必要がある

本研究では、スマートスペースおよびスマートオブジェクトへの作用に際してのこれらのような問題を改善したいと考えている。

8.2 本研究の提案

本節では、本研究の目的を述べ、それを達成すべく本研究が提案している自然言語による知的環境との対話インタフェースについて簡潔にまとめる。

8.2.1 研究目的

本研究の目的は、ユビキタスコンピューティング環境におけるスマートスペースおよびスマートオブジェクトに対しての働きかけのユーザビリティを向上することである。

8.2.2 自然言語による知的環境との対話インタフェース

本研究では、研究目的を達成するため、アプローチとして自然言語による知的環境との対話インタフェースを構築することとした。

「自然言語による対話」には以下のような特徴がある。

- 柔軟性: 同じ意味で、文面が違う表現が大量にある
- 学習コストゼロ: 日常的に使うため、新たになにかを学習する必要がない
- 意味の収束: 対話の中で、曖昧さによる文意の特定ができない場合も対話の中で段階的に明確にしていくことができる

8.3 実装

本論文の執筆にあたって、自然言語による知的環境との対話インタフェースを実現するシステム「Smart Space Conversation」を構築した。本節では、Smart Space Conversation システムの実装について簡潔にまとめる。

8.3.1 文字入力

ユーザと実際に文章のやりとりをするインタフェースとして以下の三点を実装した。ユーザはこれらのコミュニケーション手段を利用して、Smart Space Conversation システムと対話すること

ができる。

- 電子メール: 最も広く利用されているインターネット上のメッセージ交換サービス
- MSN Live メッセンジャー: 広く利用されているインスタントメッセージングサービス
- 単純なチャット: telnet コマンドから利用可能な単純な通信プロトコル

これらは実装のしやすさ、および広く使われているという点から選択したものである。単純なチャット形式は今後の開発において、他のアプリケーションから Smart Space Conversation システムと連携を行う際に使いやすいという点で実装した。

8.3.2 仮想スマートスペースモデル

仮想スマートスペースモデルは、Smart Space Conversation が制御するスマートスペースと、そこにあるスマートオブジェクトに関する情報をモデル化したデータである。今回実装した Smart Space Conversation では、運用中に動的に自身内部の仮想スマートスペースモデルを書き換えることができない。そのため、YAML 形式 [10] のスマートスペース定義ファイルを記述し、システムの設定として利用する。

仮想スマートスペースモデルでは以下のような情報を保持する。

- スマートスペースそのものに関する情報
- スマートスペース内部にあるスマートオブジェクト群
- 関連付けられたコミュニケーション手段

後述の命令認識辞書は、各スマートオブジェクトについてそれぞれ生成され、スマートオブジェクト自身によって保持される。

8.3.3 命令認識

本節では、ユーザによって入力された自然言語による表現をどう、命令として認識するかについて簡潔にまとめ、述べている。

形態素解析

まず、Smart Space Conversation では各コミュニケーション手段から入力された文字列の認識を行うための最初のステップとしてオープンソースの Java 形態素解析ライブラリ Sen [6] によって入力文字列を形態素解析を行う。これは品詞の種類や動詞の活用変化形などに応じて柔軟に認識を行うためのアプローチである。

命令認識辞書

今回実装したプロトタイプの Smart Space Conversation では、形態素の品詞に応じて柔軟な認識を行う、という実装ではなく、あらかじめ用意しておいた命令辞書と、入力された命令のパターンマッチングによって、命令認識を行っている。スマートオブジェクトの名前や、末尾の読点やクエスチョンマークの有無などの微妙な表現は許容するよう、ある程度の柔軟性を持たせた実装になっている。

対話

今回本論文の執筆にあたって実装した Smart Space Conversation では、プロトタイプであるため対話状態の状態遷移を管理する機構は実装されていない。現在ユーザがなにをやっているのか、を考慮することで命令表現の自由度と認識率を向上させることができると考えているので、対話状態遷移の管理は今後の課題の一つである。

認識できなかった場合

命令の認識に失敗した場合は、入力文中に含まれる品詞と、スマートスペース中に存在するスマートオブジェクトの持つ全ての命令と比較し、その品詞がマッチする命令を提案としてユーザに提示する。

8.4 評価

今回実装した Smart Space Conversation の評価実験を行った。被験者の人数は 10 人で、アンケートの集計結果によると過半数以上の被験者が本システムについて肯定的に評価していた。

その一方で今後の課題も評価実験を通してより明確になった。詳しくは今後の課題の節で詳述する。

8.5 今後の課題

8.5.1 認識精度向上

システムにおいて最も要求がシビアなポイントが、命令の認識精度だろう。評価実験のアンケート結果でも、認識精度の向上を求める票が過半数を超える数あり、自然言語本来の表現力には、現在の実装ではとても追いついていないとは言えないという結論が出た。

今後、命令認識モジュールを改善するにあたって、いくつかのポイントを以下に列挙する。

- 現行のルールベースの命令認識モジュールと、学習を基にした推論ベースの命令認識モジュールの精度を比べる
- 現行のルールベース命令認識モジュールのルールを拡張 (語彙追加)

- 現行のルールベース命令認識モジュールの柔軟性を上げれるポイントを探す

8.5.2 フィードバック

評価実験を通して、入力メッセージに対するシステムの返事がフィードバックとしてうまく機能していないのではないかという指摘があった。これはおそらく、命令認識に失敗した場合に大量の命令リストが送られてきてしまうため、現行の仕組みをそのまま使うのであれば、ある程度出力する命令のリストを絞り込むなど、工夫をする必要があると考えられる。もしくは、学習を元にした推論などの別のアルゴリズムを用いて関連する命令を提案する仕組みをまた実装してみるという選択肢も考えられる。

いずれにしても、実際にユーザに使ってもらって、その操作感を元に改善を地道に繰り返し、最適なフィードバック提示のあり方を模索していきたい。

8.5.3 多言語化

自然言語と名乗るからには、日本語にとどまらず、英語を初めとした外国語にも対応すべきだろう。しかし、ひとまずは日本語の実装でその基礎となる手法を固めた上で、多言語化に取り組みたい。日本語での実装でも改善すべき点が非常に多いからである。

8.5.4 音声認識技術との統合

今回実装したシステムでは、音声認識は用いず、文字入力のみで自然言語による対話とした。これには理由があり、筆者が以前オープンソースの汎用音声認識エンジン Julius [11] を使って音声認識の精度を調べてみたところ、多少のノイズがある空間ではとてもではないが使い物になるものではないと感じたためである。認識速度にも難があり、形態素に分析が完了して出力されるまでに、実際の発話からは数秒程度遅れて出力されていた。

しかし今後、音声認識技術が、問題なく日常会話を認識する精度まで進歩すれば、Smart Space Conversation システムとの結合も意味を持つだろう。

8.6 本論文のまとめ

本論文では、ユビキタスコンピューティング環境におけるスマートスペースおよびスマートオブジェクトに対する作用を、自然言語による対話的インタフェースによって行うためのシステムを提案した。スマートオブジェクトと言える高度に情報化され、ネットワーク到達性を持つ家電製品や、センサノードなどが増加してくる中、今までのリモコンや本体のボタンといった制御・情報アクセス手法ではユーザの負荷が大きい。本論文ではこれらの問題に対処するため、日常的に使う自然言語を用いた対話的なインタフェースを使うことを提案している。本論文では、自然言語を用いた知的環境との対話インタフェースのプロトタイプとして、Smart Space Conversation システム

を構築した。このシステムは評価実験の結果、過半数以上の被験者から肯定的な評価を得ることができた。また評価実験を通して、実際に自然言語を人々が日常的に使うレベルで解釈するほどにはシステムの完成度が達していないことがわかった。今後の課題として、主に命令の認識精度の向上と、対話におけるフィードバックのわかりやすさが挙げられる。

謝辞

本研究の機会を与えてくださり，絶えず丁寧なご指導を賜りました，慶應義塾大学環境情報学部教授徳田英幸博士に深く感謝致します．また，貴重なご助言を頂きました慶應義塾大学政策・メディア研究科准教授高汐一紀博士に深く感謝致します．

また，慶應義塾大学徳田研究室の諸先輩方には折に触れ貴重なご助言を頂き，また多くの議論の時間を割いて頂きました．特に政策メディア研究科修士課程徳田英隼氏には，本論文の執筆にあたってご指導を頂きました．また政策メディア研究科講師中澤仁博士，岩井将行博士には本研究を進めるにあたって多くの励ましとご指導を頂きました．ここに深い感謝の意を表します．

最後に，研究生生活を支えてくれた家族，同じ研究室の一室で家族同然に同じ時間を共に過ごした河田恭平氏，角田龍二氏，生天目直哉氏，橋爪克弥氏，また研究の日々を共に過ごした KMSF 研究グループ，HORN 研究グループの皆様に感謝の意を表し，謝辞と致します．

2008 年 2 月 7 日

伊藤 友隆

参考文献

- [1] Fedora Project. <http://fedoraproject.org/ja/>.
- [2] JINI. <http://www.sun.com/software/jini/>.
- [3] JYaml. <http://jyaml.sourceforge.net/>.
- [4] Lenovo. <http://www.lenovo.com/>.
- [5] Object Management Group. Unified modeling language. <http://www.uml.org/>.
- [6] Sen. <https://sen.dev.java.net/>.
- [7] Sun Microsystems. <http://www.sun.com/>.
- [8] The Telecooperation Office. <http://www.teco.edu/>.
- [9] Universal Plug and Play. <http://www.upnp.org/>.
- [10] YAML. <http://www.yaml.org/>.
- [11] 大語彙連続音声認識エンジン Julius. <http://julius.sourceforge.jp/>.
- [12] H. ISHII. Tangible bits: Towards seamless interfaces between people, bits and atoms. *Conference Proceedings on Human Factors in Computing Systems*, pp. 234–241, 1997.
- [13] N. KOHTAKE. u-texture : Self-organizable universal panels for creating smart surroundings. *7th International Conference Ubiquitous Computing (UbiComp 2005), September, 2005*.
- [14] Naohiko Kohtake, Takeshi Iwamoto, Genta Suzuki, Shun Aoki, Daisuke Maruyama, Takuya Kouda, Kazunori Takashio, and Hideyuki Tokuda. u-photo: A snapshot-based interaction technique for ubiquitous embedded information. In *Proceedings of The Second International Conference on Pervasive Computing (Pervasive2004) Advances in Pervasive Computing*, Vol. ISBN 3-85403-176-9, pp. 389–392, 4 2004.
- [15] T. Okoshi, S. Wakayama, Y. Sugita, S. Aoki, T. Iwamoto, J. Nakazawa, D. Furusaka, M. Iwai, and A. Kusumoto. Smart space laboratory project: Toward the next generation computing environment. *IEEE International Workshop on Networked Appliances (IWNA'01)*, pp. 115–121, 3 2001.
- [16] Mark Weiser. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, Vol. 3, No. 3, pp. 3–11, 1999.
- [17] 河口信夫. cogma : 動的ネットワーク環境における組み込み機器間の連携用ミドルウェア. 情

- 報処理学会コンピュータシステム・シンポジウム, 2001, pp. 1-8, 2001.
- [18] 俊之増井. 2000-hi-89-1 実世界 gui による情報家電プログラミング. 情報処理学会研究報告. ヒューマンインタフェース研究会報告, Vol. 2000, No. 61, pp. 1-6, 20000706.
- [19] 誠也長田, 伸一土井, 真一郎亀井. 自然言語を用いて家庭機器操作を行う対話システム. 電子情報通信学会技術研究報告. SP, 音声, Vol. 98, No. 317, pp. 23-30, 19981015.