

2005年度(平成17年度) 慶應義塾大学 政策・メディア研究科 修士論文

実空間オブジェクトに付随する
複合コンテキストの利用に関する研究

慶應義塾大学大学院 政策・メディア研究科

廣瀬 峻

tuffy@sfc.keio.ac.jp

修士論文要旨 2005 年度 (平成 17 年度)

実空間オブジェクトに付随する 複合コンテキストの利用に関する研究

-論文要旨-

本研究では、人間が日常生活の行動の中で触れるものや訪問する場所に着目し、自動個体識別技術を利用して人間の行動に関するコンテキストを表現する方法について考察した。そして、人間の行動に関するコンテキストを登録・取得・利用するシステムを構築した。

ユビキタスコンピューティングの分野において、人間の行動に関するコンテキスト情報をセンサデバイスから生成し、アプリケーションで利用するコンテキストウェア技術の研究が盛んに行われている。既存研究におけるコンテキスト情報の生成では、様々なセンサデバイスを多数用いる方法がとられている。本研究では、現状利用されているセンサで判る情報と同様に、自動個体識別技術によって得られる情報は、コンテキストを生成する上で重要であると考えられる。これは、人間の外的な活動がなんらかの物体を操作することによって行われることが多いためである。さらに、現状想定されているようなセンサネットワークは、設置コスト等の問題から早急な普及が見込めるとは言い難い。

現在、自動個体識別技術の普及が見込まれており、商品個体に一意の ID を持った RFID タグが貼付され、身の回りにある様々な物を容易に識別できるようになると考えられている。このような環境では、日常生活の中で自動個体識別が頻繁に行われ、その検出記録から人間の行動に関するコンテキスト情報の生成を行うことが可能になると予測される。複数のセンサデバイスによる検出記録を組み合わせることでコンテキスト情報を生成することが可能であるが、検出記録からコンテキスト情報を生成するシステム基盤は現在存在しない。

本研究では、検出記録群からコンテキスト判断を行うための“複合コンテキストモデル”を提案し、ミドルウェアとして設計と実装を行った。本システムを用いて、複数の検出記録を利用したコンテキスト表現を実現し、自動個体識別技術の検出記録を用いたコンテキストウェアアプリケーションを容易に構築できる環境を提供できた。

キーワード

- 1: 複合コンテキスト
- 2: コンテキスト記述
- 3: 実空間オブジェクト
- 4: 個体識別
- 5: ユビキタスコンピューティング

慶應義塾大学 政策・メディア研究科
廣瀬 峻

Abstract of Master's Thesis Academic Year 2005

A Study of Utilizing Compound Context with Real Space Objects

- Summary -

In the field of ubiquitous computing, researches on context-aware technology, in which context data on human activity generated from sensor devices is used in applications, has been a common particle. In existing researches, context data generation is achieved through usage of many sensor devices of various types. In this research, the information which can be obtained through automatic identification technology is considered as equally important as ones available through currently used sensor devices in terms of context generation. This is because human activities are often involved in dealing with objects. In addition, deployment of sensor network which is planned today is not so a simple task because of the expected installation cost.

Today, deployment of the automatic identification technology in foreseeable future is expected. This will bring an environment in which objects can be identified through RFID tags which are attached to them with unique identifiers. In this environment, it is predicted that automatic identification will often be performed in daily life, and generation of context data on human activities from detection records will be possible. Although it is possible to generate context data by combining detection records from multiple sensor devices, a system infrastructure where such a context generation is possible does not exist today.

In this research, "Compound Context Model" for estimating context from multiple detection records is designed and implemented as a middleware. This system implemented a context expression using multiple detection records, and offered an environment in which context-aware applications using detection records from automatic identification technology can easily built.

Keywords

- 1: Compound Context
- 2: Description of Context
- 3: Real Space Object
- 4: Object Identification
- 5: Ubiquitous Computing

Keio University Graduate School of Media and Governance
Shun Hirose

目次

第1章 序論	1
1.1 背景	1
1.2 研究目的	3
1.3 用語定義	4
1.4 本論文の構成	5
第2章 コンテキスト情報を利用したシステム	6
2.1 本研究が想定する環境	6
2.2 コンテキスト情報を利用したサービス	7
2.2.1 位置情報を利用したサービス	8
2.2.2 みまもりホットライン	8
2.2.3 コンテキスト情報を利用したサービスにおける問題点	10
2.3 コンテキスト情報を利用した既存研究	10
2.3.1 位置情報を用いた研究	10
2.3.2 CASP	12
2.3.3 Tagged World Project	14
2.3.4 コンテキスト情報を利用した既存研究の問題点	15
2.4 コンテキストウェアアプリケーションのためのミドルウェア	16
2.4.1 TEA	16
2.4.2 ANTH	17
2.4.3 Bazaar	17
2.4.4 コンテキストウェアアプリケーションのためのミドルウェアにおける問題点	18
2.5 既存サービス・研究の問題点	18
2.6 まとめ	19
第3章 複数のイベントによって成立するコンテキストの利用	20
3.1 EPC ネットワーク	20
3.2 RFID の利用モデル	22
3.2.1 データキャリア型 RFID システムとネットワーク型 RFID システム	22
3.2.2 ネットワーク型 RFID システムの構成要素	22
3.2.3 RFID 利用方法の検討	23
3.3 複合コンテキストモデルの提案	24
3.3.1 オブジェクトと行動	25
3.3.2 オブジェクトを利用したコンテキスト生成・表現に関する考察	26

3.4	機能要件の整理	28
3.5	まとめ	29
第 4 章	複合コンテキスト管理ミドルウェアの設計	31
4.1	設計概要	31
4.2	ミドルウェアの基本機能	33
4.2.1	イベント受信インターフェース	33
4.2.2	コンテキスト配信インターフェース	33
4.3	コンテキストツリーマネージャの設計	33
4.3.1	コンテキスト生成の条件	34
4.3.2	ツリー構造	35
4.3.3	XML による設定ファイルの記述	35
4.4	ステートマネージャの設計	35
4.4.1	ステート	36
4.4.2	ステートリスト	36
4.5	まとめ	37
第 5 章	複合コンテキスト管理ミドルウェアの実装	38
5.1	システム構成	38
5.2	センサ部	38
5.3	ミドルウェア部	41
5.4	まとめ	47
第 6 章	評価	48
6.1	評価	48
6.2	動作検証	48
6.3	機能比較	48
6.4	まとめ	49
第 7 章	結論	52
7.1	まとめ	52
7.2	今後の課題	52
	謝辞	54
	参考文献	55
	付 録 A 実験用コンテキスト設定ファイル	58

目 次

1.1	様々なセンサデバイス	2
1.2	様々な場面で利用される RFID	3
2.1	オブジェクトを用いたコンテキストの同定	7
2.2	みまもりホットライン全体概要図	9
2.3	i-POT データ表示画面	9
2.4	Active Badge	11
2.5	CASP 全体概要図	12
2.6	Tagged World Project	14
2.7	Tagged World Project で用いられるポケットアシスタント	15
2.8	Tea システムにおける階層アーキテクチャ	17
2.9	既存のコンテキスト情報利用モデル	18
3.1	EPC ネットワーク概要図	20
3.2	本研究が目指すモデル	25
3.3	時間連続性によるコンテキスト変化	27
3.4	イベントによってコンテキストが分岐する例	28
3.5	イベントのルールセット	29
4.1	ミドルウェア構成図	32
4.2	順列	34
4.3	組み合わせ	34
4.4	順列と組み合わせ	35
4.5	ステート	36
5.1	システム構成	38
5.2	RFID タグ	39
5.3	携帯型 RFID リーダ端末	40
5.4	リーダから送られてくるイベントの構造体	40
5.5	XML ファイルパーサ関数	42
5.6	コンテキストツリー例	42
5.7	コンテキスト設定ファイル例	43
5.8	ステート管理構造体	44
5.9	コンテキスト検索関数	45
5.10	フローチャート図	46

6.1	コンテキスト検索	50
6.2	検出イベントログ	51

表 目 次

2.1	Active Badge System で提供される API	11
2.2	コンテクストライブラリの例	13
2.3	コンテクスト情報の例	13
2.4	既存研究の比較	15
2.5	既存ミドルウェア比較	18
3.1	EPC コード体系	21
3.2	RFID 利用モデルの比較	24
5.1	CF カード型 RFID ユニット V720S-HMF01 仕様	39
5.2	Fujitsu Pocket Loox FLX3AW 仕様, および実装言語	40
5.3	実装環境	41
6.1	既存ミドルウェアと本システムとの比較	49

第1章 序論

本章では、本研究の背景、および目的について述べる。また、本論文の構成を示す。

1.1 背景

ユビキタスコンピューティング環境

近年、ユビキタスコンピューティング [1] という言葉が社会一般に認知されるようになった。ユビキタスコンピューティングとは、社会や生活の至る所にコンピュータや入出力デバイスが遍在 (ubiquitous) し、コンピュータ同士が自律的に連携して動作することで、我々ユーザに対して多様なサービスを提供するといった考え方を指すもので、1991 年にアメリカの Mark Weiser によって提唱された概念である。ユビキタスコンピューティング環境では、PC や PDA(Personal Digital Assistance) といったコンピュータ端末に限らず、自動車や家電製品、温度計、照度計などさまざまな機器が計算資源やネットワーク接続性を持ち、協調動作することによって多種多様なサービスを実現することが見込まれている。実際にインターネット自動車や情報家電といった形で現在研究・開発が進んでおり [2]、その一部は既に製品化されている。ユビキタスコンピューティング環境におけるサービスの具体例として、ユーザの位置情報に基づいて、その時その場所で最適な情報をユーザに提供するサービスや、ユーザの動きを検知することで部屋の明るさや室温を自動的に調節するサービスなどが挙げられる。

コンテキストアウェアネス

ユビキタスコンピューティング環境を語る上で不可欠な考え方として、コンテキストアウェアネスが挙げられる。コンテキストアウェアネスは、context-awareness というように context と awareness の二つの単語から成り立った言葉である。この二つの単語を辞書で引くと、下記の通り記述されている。

context

1. the situation, events, or information that are related to something and that help you to understand it
2. the words that come just before and after a word or sentence and that help you understand its meaning

awareness

1. knowledge or understanding of a particular subject or situation

2. the ability to notice something using your senses

(Longman Language Activator より)

すなわち、コンテキストウェアネスとは一般的に、“ 周囲にある何かに関する状況や情報を認知すること、していること ”と言える。ただし、これは非常に抽象度の高い言葉であり、その解釈は言葉を用いる各人によって差異がある。1994 年に文献 [3] で初めて context-aware という言葉が定義されたが、特に最近ではセンサデバイスの小型化、高機能化が進み、図 1.1 に示したような温度計や湿度計、赤外センサや GPS(Global Positioning System) など様々なデバイスを用いて人間の周辺環境情報を取得できるようになったことから、コンテキストを一つの意味で共通化するのは非常に困難である。ただし、コンテキストウェアネスの研究に共通する目的は、ある時点での人間の周辺環境状況を理解し、その時、その場所において最も最適な選択を提供することによって“ 人間の行動を支援する ”ことであると言える。

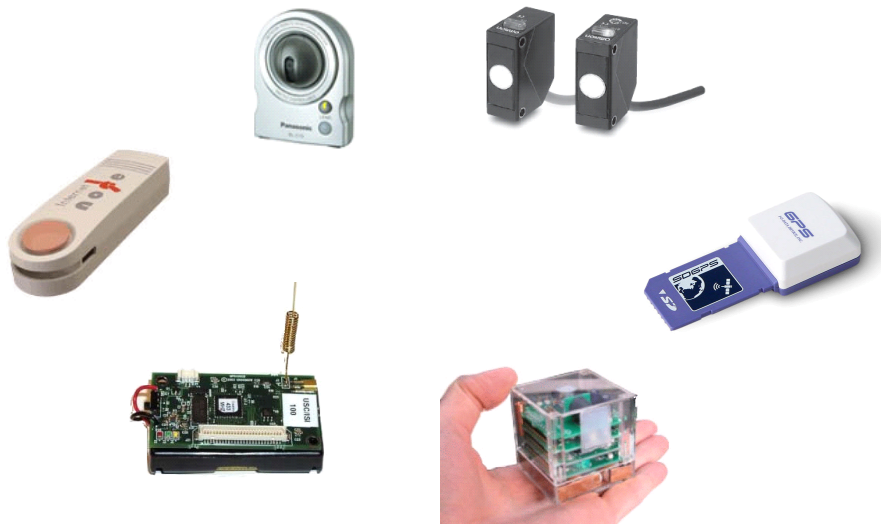


図 1.1: 様々なセンサデバイス

RFID の社会的普及

近年、RFID(Radio Frequency IDentification) 技術を用いた自動個体認識技術が広く展開しつつある。我々の身の回りを見渡すだけでも、図 1.2 に示すとおり Sony が開発した FeliCa[4] を利用したサービスである電子マネー Edy[5] や、JR の改札に採用されている Suica[6]、Icoca[7] などによりその技術が使われており、非接触型 IC カードとして広く利用されている。これにより我々利用者は、財布や定期入れからお金や定期を出さずに財布や定期入れをそのまま読み取り端末にかざすだけでスムーズに決済を行ったり、改札を通ったりすることができるようになった。この他に、RFID を用いた例として、空港での航空手荷物管理システム [8] や、病院において薬品と投与患者のマッチングを行うことで薬品投与ミスや患者取り違いなどの医療事故を未然に防ぐためのシステムなどが開発され、現在導入が進められている。

また、近年の社会的動向として、一意な識別子を保持した RFID タグを製品につけることで、流通過程での盗難防止や在庫管理の効率化を行う取り組みが多々見受けられる。これは、

SCM(Supply Chain Management) で発生するコストの大幅な削減, および製品トレーサビリティを実現する目的で, アメリカの WalMart を中心とした組織 EPCglobal[9] などが進められている. このように, RFID を用いた新しいシステムに対する期待は大きく, 数年後には我々が買った製品一つ一つに現在の製品種類単位での識別を可能にしてきたバーコードではなく, 個体レベルで製品を識別できる RFID が貼付された世界が来ることが予想されている.

また, そういった環境を想定し, 製品個々のメタ情報をネットワーク上で管理するための研究活動を Auto-ID Labs[10] が中心となって行っている. その研究範囲はタグの読み取り率の向上といったハードウェアの分野から, RFID システムアーキテクチャの提案などのソフトウェア分野まで多岐にわたっている.



図 1.2: 様々な場面で利用される RFID

1.2 研究目的

本研究では, 人間が日常生活における行動の中で触れるものや訪問する場所に着目し, 自動個体識別技術を利用して人間の行動に関するコンテキストを表現する方法について考察する. そして, 人間の行動に関するコンテキストを容易に登録・取得・利用するためのシステム構築を行った.

本研究では, 以下の 2 点に注目し, システムの設計・実装を行った.

1. オブジェクト検出情報からのコンテキスト抽象化
オブジェクトの検出という単純なセンサ情報から, システムが理解可能な形でコンテキストを表現する手法について考察した.
2. 複数センサイベントの複合的な利用
これまで, ある一時点でのセンサ情報を使って判断されることが多かったコンテキストを, ある一時点の前後にあるセンサ情報などと組み合わせて判断する必要性について議論した.

1.3 用語定義

本節では、本研究で用いる用語を定義する。

オブジェクト

本研究ではモノや製品、場所などに RFID タグが貼付された環境を想定する。本稿では、このタグが貼りつけされたモノや製品、場所などを総称してオブジェクト、あるいは実空間オブジェクトと定義する。

コンテキスト

“ コンテキスト ” という言葉を辞書でひくと “ 文章などの前後の関係. 文脈. ” とある (大辞林 第二版)。しかし, “ コンテキスト = 文脈 ” と一言でいっても, 実際にコンテキストという言葉を使う場合には様々な意味で用いられることが多い。そこで, 本節ではコンテキストという言葉について整理し, 本研究におけるコンテキストという言葉の定義を行う。

1.1 節で述べたように, “ コンテキスト (context) ” という言葉は 1994 年に文献 [3] において初めて定義された。そこでは分散モバイルコンピューティング環境を前提として下記のよう

Three important aspects of context are: **where you are, who you are with, and what resources are nearby**. Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social situation; e.g., whether you are with your manager or with a co-worker.

すなわち, 分散モバイルコンピューティング環境におけるコンテキストは, その人が現在いる場所, 現在その人と一緒にいる人, 現在その人の近くにあるもの, という 3 つの要素で構成されると述べている。

また, コンテキストウェアネスに関連する論文において, 文献 [3] と並んで参照される Dey らの論文 [11] では, 以下のようにコンテキストを定義している。

Context : any information that can be used to characterize the situation of entities (i.e. whether a person, place or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, groups and computational and physical objects.

(コンテキスト:ある実体の状況の特徴づけることができる様々な情報である。実体とは, ユーザとアプリケーション間のインタラクションに関する人や場所, 物であり, ユーザやアプリケーション自体も含まれる。すなわち, コンテキストとは場所や識別子, さらに個人, グループ, コンピュータや実空間オブジェクトなどの状態を表すものである。)

このように、コンテキストウェアネスの分野における代表的な論文同士ですらコンテキストの定義にはそれぞれ若干の差異がある。上述した両者に共通する部分を、“実空間の中に存在するなんらかの状態を表す情報”，ではないかと考えた上で、本研究におけるコンテキストを下記のように定義する。

コンテキスト = あるオブジェクトを識別することによって得られる、人間の行動に関して意味付けされた情報

イベント

コンテキストウェアアプリケーションでは、前節でも述べたように様々なセンサデバイスが利用される。そして、これらのセンサデバイスでは、機器ごとに様々な粒度で様々な情報を取得している。本研究では、センサがある一時点でセンシング可能な情報を取得し、データを情報収集サーバに送るという一連の動作をイベントとして定義する。このイベントは、本研究におけるコンテキストを構成するための最小構成単位である。

トリガ

コンテキストウェアアプリケーションにおいて、あるイベントの検知、あるいは状態の変化をもとにアプリケーションが駆動することが多い。本研究では、このアプリケーションに対して駆動を促す情報をトリガと定義する。

1.4 本論文の構成

本論文は以下の 6 章により構成される。

第 2 章ではコンテキストウェア分野における既存の研究・サービスについて考察し、本研究における問題意識について述べる。第 3 章では、その問題を解決するための機能要件、およびアプローチについて考察し、複合コンテキストモデルの提案を行う。第 4 章では、アプローチに基づいて、複合コンテキストを利用するためのシステム設計について述べる。第 5 章では、システムの具体的な実装について述べる。第 6 章では、実装したシステムに対して行った評価と動作検証について述べる。第 7 章で、まとめと今後の課題について述べ、本論文の結論とする。

第2章 コンテキスト情報を利用したシステム

本章では、まず本研究が想定している環境について述べる。具体的には、実空間上に存在する様々なオブジェクトに付随したコンテキスト情報を柔軟に利用できる環境である。その後、想定する環境を実現していくにあたり、関連するコンテキスト情報を利用した既存のシステムやサービスについて述べ、問題点について整理を行う。

2.1 本研究が想定する環境

1.1 節で述べた通り、ユビキタスコンピューティングの概念が社会に広く浸透し、RFID が身の回りの様々な製品や場所に貼付されているのが当たり前な環境に近い将来やってくると考えられる。本研究では、RFID が様々なオブジェクトに貼付され、人間が RFID リーダを持ち歩く環境を想定する。その環境において、RFID リーダから取得されるイベント情報をコンテキスト情報へと変換し、人間に対して何らかの形で行動支援を行うことを考える。例として、実空間オブジェクトを利用してコンテキストを生成する場合を考える。今後普及が見込まれる商品用 RFID を利用することにより、従来のセンシングデバイスなどを用いた手法に比べて、より多くの実空間オブジェクトを識別・監視することが可能となる。

RFID のみをセンシングデバイスとして用いる場合と、RFID に限定せず他の多様なセンサデバイス群を用いた場合を比較すると、後者の方が多様な状況認識を行うことができ [12]、提供可能なサービスの幅は広がると考えられる。しかし、カメラなどの光学センサや赤外線センサを用いても、オブジェクトの存在自体は認識できるが、その認識したオブジェクトが“何であるか”を識別することは非常に困難である。また、身の回り全てのオブジェクトに ID を登録することは事実上不可能である。EPCglobal が展開する EPC(Electronic Product Code)[13] をはじめとした、流通に利用される商品識別用の RFID タグは、個品情報を示すタグとして利用される。そこで、この EPC を利用することで、身の回りに存在するオブジェクトを容易に識別することが可能になると想定できる。また、多様なセンサデバイスをユーザの周辺環境に組み込むためには、高い設置コストが必要となる。

よって、本研究では RFID を用いて、人間の行動支援を行うことを考える。

次に、RFID を用いて家庭環境において行動支援を行う例として、図 2.1 を用いて説明する。図 2.1 は家庭環境における一般的な場面を切り出したものである。

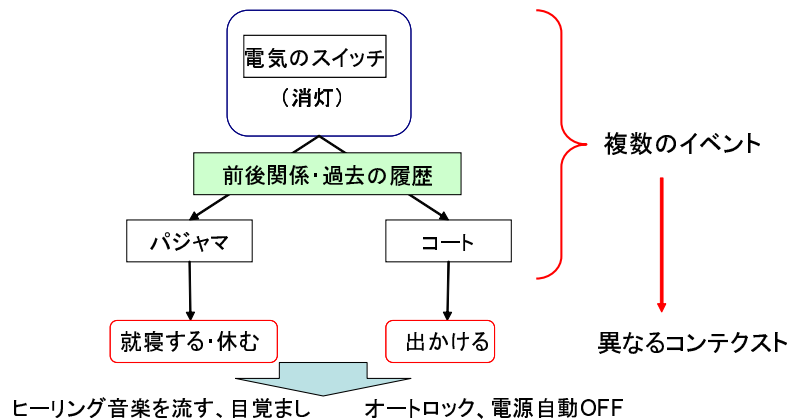


図 2.1: オブジェクトを用いたコンテキストの同定

図 2.1 の例では、電源スイッチというオブジェクトを取得することによって電源が OFF になったという情報を取得したと仮定する。このイベント情報は、一つのコンテキスト情報として利用可能である。しかし、人間の日常生活を考えた場合、あるオブジェクトのコンテキスト情報は、オブジェクト検出の時間や、オブジェクト検出の前後に検出した別のオブジェクトとの関係性によって大きく変化する可能性がある。また、コンテキスト情報を利用したサービスを考えた場合、サービス利用者やオブジェクトの検出場所によって同じオブジェクトに対して意識するコンテキストは異なる可能性がある。例えば、電源スイッチを検出した前後にパジャマというオブジェクトが検出された場合、それは“ 就寝する・休む ”というコンテキストが存在する可能性がある。また、前後に検出したオブジェクトがパジャマではなく、外出用の上着である場合、それは“ 出かける ”というコンテキストである可能性がある。

このように、人間の行動に関するコンテキストが取得できれば、個人の行動をサポートするアプリケーションを構築することが可能である。

上述した例のように、実空間上のオブジェクトに付随するコンテキスト情報を利用してサービスを提供することは、近い将来において実現される可能性が高い。本章の以降の節では、そのような環境を考えた時に関連する既存のサービスや研究、システム基盤などについて考察することで、どのような課題が存在するのか整理する。

2.2 コンテキスト情報を利用したサービス

本節では、現在実際に提供されているコンテキスト情報を利用した情報サービスについて述べる。

コンテキスト情報を利用したサービスは、基本的にユーザ (サービスを受ける人) の周辺環境情報をセンサから取得する必要がある。サービスとして提供するためには最初に基盤の整備が必要不可欠である。そのため、現在サービスとして社会に提供されているシステムは、世間一般に広く普及している携帯電話やカーナビゲーションシステムについた GPS を用いて取得可能な位置情報を利用するものが大半を占めている。

2.2.1 位置情報を利用したサービス

現在、携帯電話が接続している基地局や、GPS から得られる位置情報を利用したサービスが数多く提供されている。本節では、それらの中から代表的なサービスについて説明する。

携帯電話を利用した位置情報サービス

携帯電話を利用した位置情報サービスの例として、i エリアがある。i エリアは、NTT ドコモの i モード対応電話機で利用できる位置情報サービスである。i エリアは、携帯電話が接続している基地局から現在地を取得し、全国を 500 ヶ所程度に分割したエリアごとの情報を受け取ることができる。エリアごとの情報提供が可能のため、現在各エリアに特化した天気予報、地図、交通情報、グルメや買い物情報などが提供されている。同様のサービスとして、au の EZWeb サービスの一部である GPS 位置情報サービス “eznavigation” や、vodafone の地図情報サービス “Vodafone Map” などがある。これらのサービスは、各キャリア毎に位置情報の取得方法は異なるが、利用者の現在地をコンテキストとして解釈する点で共通している。そして、このコンテキストにもとづいて、現在地周辺の情報など、利用者にとって “その時、その場所で” 価値の高い情報を提供している。

丸の内ユビキタスミュージアム

丸の内ユビキタスミュージアム [14] は、ユーザの位置情報を GPS で取得するのではなく、環境側に設置された 2 次元バーコードの QR コードを、ユーザが携帯カメラなどを用いて読み込むことにより、読み込んだ QR コードの設置場所を現在位置として取得する。そして、その場所近辺で行われているイベント情報を提供したり、掲示板機能を提供したりするなど、その場所に行かないと得られない情報をユーザに提供している。

2.2.2 みまもりホットライン

みまもりホットライン [15] とは、象印マホービン株式会社が提供しているサービスで、一人暮らしの老人が毎日元気に過ごしていることを遠方に住む家族に伝えるためのサービスである。みまもりホットラインでは、元気なお年寄りが毎日使うであろう機器の中でも特にポットに着目した。ポットに着目した理由としては下記の通りである。

- 毎日使う機器の中でも特に利用者毎に使い方、すなわちお茶を入れる時間のパターンがある程度一定である【認知容易性】
- どこの家にもあり、サービス導入による圧迫感がない【心的負担の軽減】
- ボタン一つ押すだけで使える【導入容易性】

みまもりホットラインのサービス全体概要を示したのが図 2.2 である。

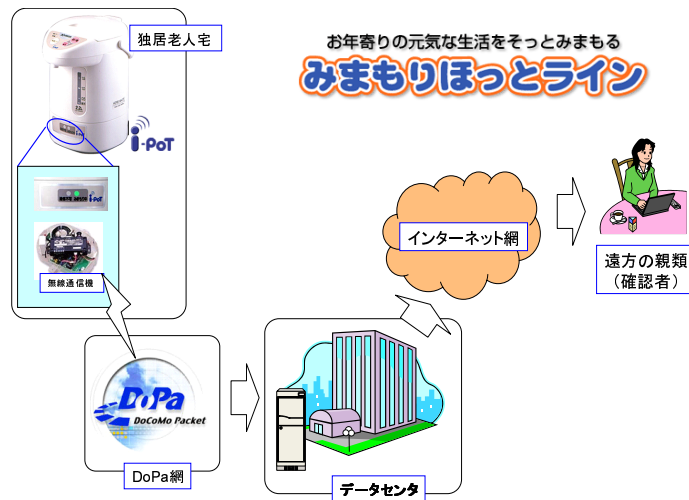


図 2.2: みまもりホットライン全体概要図

このサービスでは、ポットの電源を ON/OFF したり、給湯したりするたびに、内蔵されている無線通信機からポットの操作信号が発信され、そのデータが DoPa[16] 網を通してデータセンタの専用サーバへ送られる。そして、遠方に住む家族は、そのデータをもとにした使用状況を 1 日 2 回 E メールで受け取れるほか、専用ホームページや携帯電話から直近の数日の利用データを図 2.3(みまもりホットライン HP より一部利用) に示したようなグラフで確認することができる。これにより、一人暮らしの老人と遠方に住む家族や知人双方に大きな負担を強いることなく、遠方の家族は老人の健康状態を把握することができる。

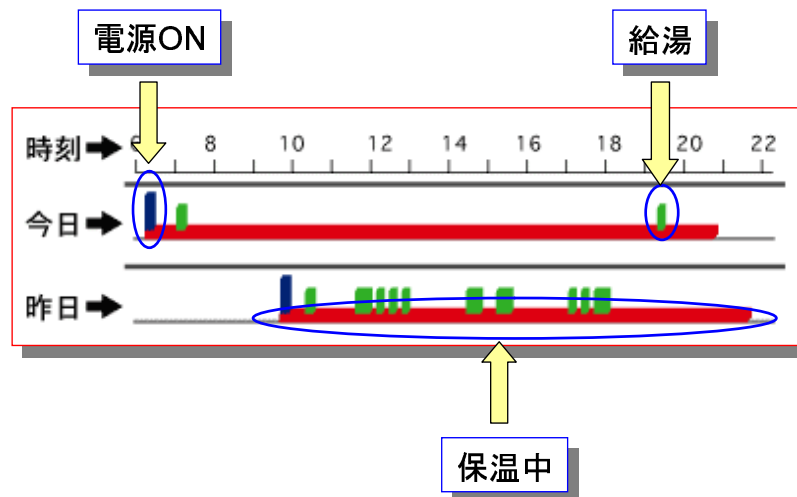


図 2.3: i-POT データ表示画面

次に、このサービスにおけるコンテキストについて考察する。みまもりホットラインでは、電源の ON/OFF、給湯という各々のイベントに対して、一人暮らしの老人が“今日も元気に生活している”というコンテキスト情報が付加されている。

2.2.3 コンテキスト情報を利用したサービスにおける問題点

コンテキスト情報を利用した代表的なサービスについて述べたが、これらのサービスでは、センサから取得された情報がそれぞれ個別に利用されており、一つのセンサイベント情報が別のセンサイベント情報と組み合わせて利用されることは想定されていない。すなわち、位置情報サービスの例であれば、ある時点での位置情報に対して、それまでにユーザが通って来た位置情報と組み合わせてユーザのコンテキストを判断し、サービスを変えることは想定されていない。

2.3 コンテキスト情報を利用した既存研究

本節では、コンテキストアウェアネスの分野における既存研究について、想定する環境を念頭に置きながら述べる。

2.3.1 位置情報を用いた研究

我々の生活は位置情報に依存することが多い。そのため、コンテキストアウェアネスの研究分野でも、位置情報は早くから重要なコンテキスト情報として考えられ、様々な研究が行われてきた。例えばセンサネットワークで温度や湿度といった情報を取得する際にも、それがどの場所の情報なのかを把握する必要がある。最近では特に GPS の急速な普及を背景として、移動する人間の位置情報に応じて適切なサービス提供を行うための研究が多々為されている。屋内での位置情報利用を想定したサービスとして C-MAP[17] や赤外線センサを用いた Active Badge[18][19]、超音波センサを用いた Active Bat[20]、Cricket[21][22] などが挙げられる。また、屋外での位置情報利用を想定したサービスとして Cyberguide[23] や GUIDE[24] などが挙げられる。これらのシステムは、一般的にロケーションアウェアシステムとして分類されることが多い。

Active Badge System

Active Badge System は AT&T 研究所で提案されたシステムで、赤外線を発信する Active Badge を人やモノなどに備えることによって、屋内における人やモノの位置情報を取得する。人の位置や状態に応じて挙動を変えるコンテキストアウェアアプリケーションの実現を目的としているため、正確で詳細な位置情報の認識や、環境認識が可能である。しかし、このシステムは、敷設にコストがかかるため容易に導入できないという欠点がある。また、ある場所におけるセンサの検出イベントが、他の場所あるいは他の時間に検出された別の検出イベントと連携して何かをサービスするということは想定されていない。Active Badge System のデモンストレーションシステムでは、位置情報システムへのアプリケーションインタフェースとして表 2.1 に示すコマンド群が用意されている。センサ情報の記述においては CORBA オブジェクトを利用しており、アプリケーションに対して CORBA オブジェクトによるセンサ情報の提供を行う。



図 2.4: Active Badge

表 2.1: Active Badge System で提供される API

コマンド	概要
FIND(name)	name の badge が存在する場所を通知
WITH(name)	name の badge が存在する場所およびその場所に存在する他の badge の情報を通知
LOOK(location)	location の近傍に存在する badge の情報を提供
NOTIFY(name)	name が次に Active Badge System で観測された時に通知
HISTORY(name)	name の位置を一時間周期で報告

Active Bat

Active Bat も Active Badge 同様，AT&T 研究所で提案されたシステムであり，赤外線センサを用いていた Active Badge とは異なり，超音波センサを利用することにより数センチメートル単位での検出精度を実現している．

Cricket

Cricket は，無線電波と超音波を利用した位置検出システムである．Cricket beacon と呼ばれる機器を建物内に複数配置し，無線電波と超音波を定期的に発信する．Cricket compass と呼ばれる機器を小型機器につけ，無線電波を受信してから超音波を受信するまでの時間差によって compass の位置を取得する．また，超音波センサを複数用いることにより，向き情報も取得できる．想定アプリケーションとして，Cricket compass から距離と方向を指定し，その範囲内にあるデバイス一覧を取得する ViewFinder や，位置情報と向き情報を利用して，ユーザに建物内の道案内をする WayFinder などがあげられる．

Cyberguide, GUIDE

Cyberguide と GUIDE は、共にユーザの現在地に関する情報を提供したり、興味のあるような場所を提案したりするシステムである。Cyberguide は GPS を用いて、また、GUIDE は IEEE802.11b の無線 LAN 基地局を用いてユーザの位置を把握する。

2.3.2 CASP

CASP とは Context Aware Service Platform の略で、KDDI 研究所で研究されている RFID を用いたサービスプラットフォームである [25]。この研究では、あらゆるオブジェクトに物流などで利用される RFID が貼付され、一意に識別できるような環境を想定したときに、タグ ID 情報と人間の行動との対応づけ、およびその情報を利用して人間の行動を支援することに主眼を置いている。すなわち、人間の行動をコンテキストとしてとらえることによってシステムに組み込んでいる。具体的には、“時刻属性、場所属性を基本属性とし、その他いくつかの属性から構成される情報”をコンテキストとして定義している。

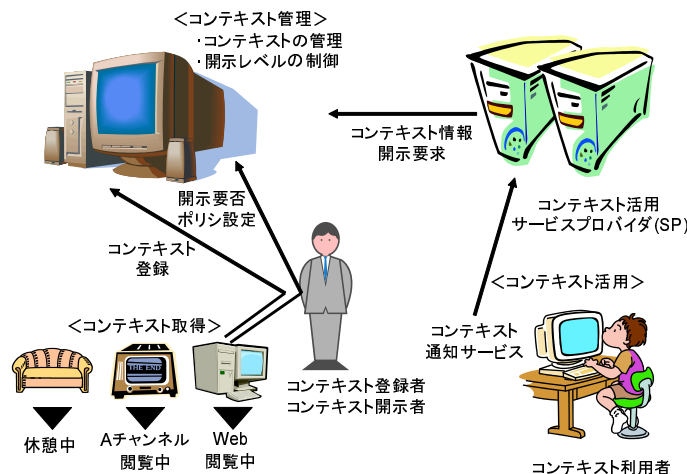


図 2.5: CASP 全体概要図

図 2.5 に CASP の全体概要を示す。CASP は主に 4 つの部分から構成されている。

1. コンテキスト登録者
オブジェクトに関するコンテキスト情報を登録する人。
2. コンテキスト管理サーバ
後述するコンテキストライブラリとコンテキストリストを保持したサーバ。
3. コンテキスト活用サービスプロバイダ
コンテキスト情報を活用するサービス提供者であり、複数ユーザ間でコンテキスト情報を通知するなどのサービスを提供する。
4. コンテキスト利用者
コンテキスト情報を利用する人。コンテキスト登録者以外に、第三者も含まれる。

コンテキストの登録

CASP では，表 2.2 に示したように，オブジェクトに貼り付けされたタグのタグ ID 情報と一意に対応付けられたオブジェクト名に対して，コンテキストの候補が予め登録されている．なお，一つのオブジェクトに対して複数のコンテキスト候補が登録されている場合もある．このオブジェクト名とコンテキストとの対応の集合をコンテキストライブラリと呼んでいる．このコンテキストライブラリに登録されている複数候補の中から，適切なコンテキストをコンテキスト登録者が選択し，サーバに登録する．

表 2.2: コンテキストライブラリの例

オブジェクト名	コンテキストライブラリ (コンテキストの候補)
A 電鉄 B 駅改札口	A 電鉄に乗車する A 電鉄を降車する
オフィス在籍表	出社する 退社する 外出する
パソコン	メールを書く WEB を見る
ソファ	休憩する

次に CASP における，コンテキスト情報の登録について説明する．CASP では，リーダを各コンテキスト登録者が所持し，あらゆる場面でコンテキスト登録者が行動すると同時に，行動時に触れたオブジェクトのタグ ID 情報が取得できることを想定している．そして，取得したタグ ID をもとに，そのオブジェクト情報を取得し，表 2.3 に示したコンテキスト記述フォーマットに従ってサーバ上にコンテキスト情報を蓄積していく．なお，このコンテキスト情報の集合をコンテキストリストと呼んでいる．そして，これらコンテキストライブラリとコンテキストリストをコンテキスト管理サーバで管理している．

表 2.3: コンテキスト情報の例

time 属性	place 属性	object 属性	context 属性	開示レベル
2002-12-16-20:23	自宅	パソコン	ゲームをした	2

コンテキストの利用

コンテキスト利用者がコンテキストを利用する際には，図 2.5 にあるコンテキスト活用サービスプロバイダ (SP) を経由してコンテキストリストを取得する．その際，表 2.3 にある“開示レベル”によってコンテキスト利用者に対して簡単なアクセス制御を行っている．

以上，CASP について概説したが，CASP では一つ一つのイベントに対応するコンテキス

ト情報を時系列に沿って登録・管理しており、複数のイベントを考慮したコンテキスト管理方法について考慮されていない。

2.3.3 Tagged World Project

Tagged World Project[26] では、RFID を利用した行動支援サービスの実現を目的としている。我々の身の回りにあるあらゆるものに RFID タグを貼付することで、RFID リーダを携帯した人間が、普段の行動の中でこういったものに触れているのかを取得・蓄積する。そして、この蓄積されたデータをもとにユーザの行動パターンを計算機上に表現し、ユーザの行動を推定することで、人間の行動を支援してくれるサービスの実現を目的としており、現在立命館大学の島川研究室や京都高度技術研究所などが参加して研究が行われている。

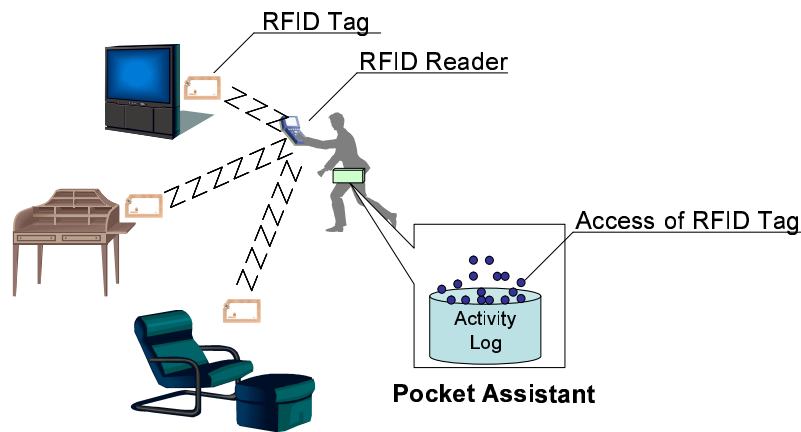


図 2.6: Tagged World Project

Tagged World Project のイメージを図 2.6 に示す。Tagged World では、オフィスや学校、店舗など、ユーザが日常的に繰り返し訪れ、さらに時間的・物理的に区切られている空間をサービスの対象としている。その中で、各ユーザは図 2.7(文献 [27] より) に示すポケットアシスタントという RFID リーダが接続された携帯型小型サーバを持ち、このポケットアシスタントにユーザが行動の中で触ったものの ID とその名前を蓄積していく。そして、個人が所持するポケットアシスタントには、特定状況での行動の習慣や癖をパターン化したデータが保持されており、そのパターンと実際に取得されたデータのパターンとを比較することによってユーザの行動を推定する。また、個人が所持するポケットアシスタントにデータを蓄積することによって、従来のユビキタス基盤で実現されていた個人情報の管理方式と比較してプライバシー管理が容易となるとも述べている。



図 2.7: Tagged World Project で用いられるポケットアシスタント

この研究では，RFID を用いてユーザの行動を認識するにあたり三段階の抽象化を行っており [28]，本研究で RFID タグからの情報取得モデルについて考える際の参考となる．

以上述べてきたように，Tagged World Project は，我々人間の周辺環境に存在する様々なオブジェクトに RFID が貼り付けられた環境を想定し，RFID を利用して人間の行動認識を行うことで行動支援を行うことに主眼をおいており，本研究を進めるにあたって参考となる部分は多い．しかし，アプリケーションオリエンテッドなアプローチになっており，汎用的にこのシステムを利用していくのは困難である．

2.3.4 コンテキスト情報を利用した既存研究の問題点

表 2.4 に，これまで述べてきた既存研究についてまとめる．

この表からも明らかだが，既存研究では，複数イベントを考慮し，かつ拡張性を有したシステムになっていない．

表 2.4: 既存研究の比較

	ロケーションウェアシステム	CASP	Tagged World
コスト	高い	安い	安い
複数イベントの考慮	×	×	△
実空間オブジェクトへの透過性	△	○	○
拡張性	△	△	×
時系列概念	×	×	○

各比較項目について以下に詳述する．

コスト

実際にシステムを導入する際にかかる設置コストが高いか安いのか

複数イベントの考慮

センサから取得されたイベント情報を扱う際、複数のイベントをひとつのコンテキスト情報として扱えるか

実空間オブジェクトへの透過性

システムが実空間オブジェクトを扱える設計になっているか

拡張性

システムを拡張する際に、容易に拡張ができるか

時系列概念

センサから取得されたイベント情報を、ある一時点での情報でしか扱えないのか、またはある一定の時間幅の中でイベント情報を扱えるのか

2.4 コンテキストウェアアプリケーションのためのミドルウェア

コンテキストウェアアプリケーションの現状は、様々なセンサデバイスが存在することによって多様性が発生するため、アプリケーション毎に何もない状態から開発するか、必要な機能を高次元のレベルで抽象化し、ミドルウェアとして提供されたものを利用するかといういずれかにあてはまる場合がほとんどである。今後、個体識別技術が普及することにより、様々なアプリケーションが開発されることが予想され、ミドルウェアはシステムの中でも重要な位置を占める。

そこで、本節ではコンテキストウェアアプリケーションの分野におけるミドルウェアについて考察する。

2.4.1 TEA

TEA システム [29] は、図 2.8(文献 [29] より) に示す通り、実空間の情報、コンテキストをレイヤリングモデルで表現している。各センサ毎の情報を“ Cue ”レイヤで収集、表現する。複数の Cue を組みあわせることにより、上位の“ Context ”レイヤでより複雑な実空間の情報を表現することができると述べており、本研究と類似した概念にもとづいている。TEA システムは、レイヤリングモデルを採用することで、多種多様なセンサデバイスに対応可能である。しかし、これらのセンサデバイスから取得した情報をどのように管理するかについて述べられておらず、他のアプリケーションに対して汎用的に利用することは困難である。

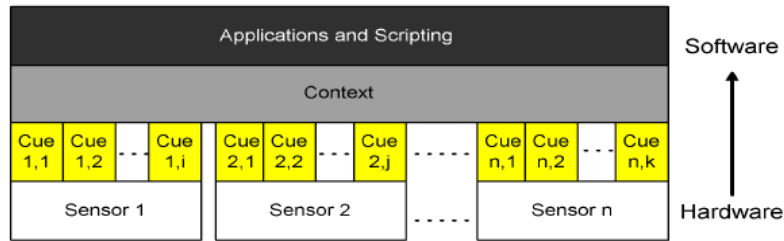


図 2.8: Tea システムにおける階層アーキテクチャ

2.4.2 ANTH

ANTH(ANTennary THings)[30] は、ライトや時計、ペンなど我々ユーザの身の回りに存在する単機能的なオブジェクトに注目し、それらのオブジェクトがネットワークを介して相互に接続することにより構築される空間を、“空間としてのコンピュータ”としてとらえている。ANTH は、実空間オブジェクトの機能同士を連携させることで提供できるサービスを、ユーザがプログラムするための実空間プログラミングフレームワークである。

オブジェクト同士の連携を考える上で、Jini[31] や UPnP[32] が有名であるが、これらの技術は電源や計算資源などがある程度備わっている情報家電を対象としている。ANTH で対象としているのは家庭環境の中でも特に単機能的なオブジェクトであり、そういった資源に恵まれていない。よって、連携させるオブジェクトの機能を絞り込むことで簡潔に抽象化し、機能間の関係を単純に記述可能にすることを目標としている。

ANTH では、例えば「目覚し時計のタイマが機能したら (event)、コーヒーマーカの電源が ON になる (action)」というように、目覚しとコーヒーマーカそれぞれの単機能を event と action、すなわち“条件と動作”で表現することでオブジェクト間連携を行う。event と action は、本研究におけるイベントとコンテキストの関係に似ているが、ANTH では ANTH チップという一つのハードウェアモジュールとして機能が提供されており、この ANTH チップ同士がアドホック通信を行うなど実運用を考えた場合、実空間上のオブジェクト全てにこのチップを付けることは非常に困難であると考えられる。

2.4.3 Bazaar

Bazaar[33] では、なんらかの識別子 (ID) が付与された日常物について、その存在場所と ID に対応した属性情報を関連づけて管理する。そして、画像センサや光センサ、圧力センサなどの様々なセンサデバイスを用いて、日常物の状態を把握したり、アプリケーションにコマンドを発呼するなどの手段を、抽象化されたクラスとしてアプリケーション開発者に提供している。

管理対象となる日常物のメタ情報を主として製造業者が提供することや、IDResolver と呼ばれる名前解決コンポーネントが存在することを前提にするなど、本研究が前提としている EPC ネットワークと重なる部分は多い。しかし、オブジェクトの検出という単純なイベントのみを用いてコンテキストを生成することは考えられていない。

2.4.4 コンテキストウェアアプリケーションのためのミドルウェアにおける問題点

本節では，コンテキストウェア分野の中でも，特にミドルウェアの部分に着目して考察を行った．表 2.5 に，考察した結果をまとめる．

表 2.5: 既存ミドルウェア比較

	TEA	ANTH	Bazzar
複数イベントの考慮	○	△	×
実空間オブジェクトへの透過性	△	×	△
時系列概念	△	△	△

このように，実空間オブジェクトを扱うことを考慮したとき，既存のミドルウェアでは実空間オブジェクトの特性に対応しきれておらず，本研究で想定する環境に対応できない．

2.5 既存サービス・研究の問題点

これまで述べてきたように，現在行われているコンテキストウェアネスの研究分野では，様々なイベントを単独のトリガとして利用しているものが多い．すなわち，GPS を利用した位置履歴表示システムや，現在いる場所の位置情報を利用した提案型ナビゲーションシステムなどでは，センサから取得される一つ一つのイベントがそれぞれ独立して利用されている．(図 2.9)

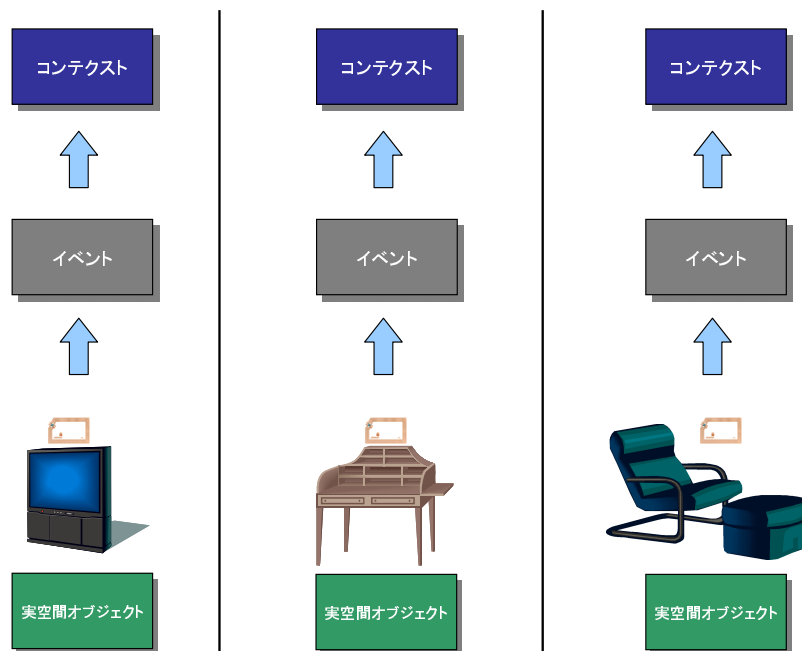


図 2.9: 既存のコンテキスト情報利用モデル

RFID の普及により様々なオブジェクトに識別子が付き、あらゆるオブジェクトを一意に識別できるようになると、オブジェクトに付随する様々なコンテキスト情報が発生し、それを利用できるようになる。しかし、オブジェクトに付随するコンテキストは、検出されたオブジェクトの前後関係や周辺環境、さらにコンテキストを利用する各個人によって大きく変化し得る。既存のミドルウェアは、オブジェクトを対象としてシステムが扱えること、およびアプリケーション開発の際に重要な汎用性の双方を兼ね備えた設計になっていない。

実空間オブジェクトを用いた人間の行動支援を考えると、複数のイベント情報によってコンテキストが決定されることが多いため、複数のセンサイベント間の関係性を考慮したシステム基盤が必要となる。

2.6 まとめ

本章では、コンテキストに着目した既存の研究およびサービスについて、イベントとコンテキスト利用の関係という観点から問題を整理した。また、コンテキストウェアアプリケーションを開発する上で重要となるミドルウェアについて既存の研究を挙げ、考察を行った。

次章では、複数のイベントを利用して判断されるコンテキストの利用について考察していく。

第3章 複数のイベントによって成立するコンテキストの利用

本章では、まず初めに本研究が前提としている EPC ネットワークについて概説し、RFID の利用モデルについて考察を行う。その後、第2章で整理した問題を踏まえ、問題解決のための機能要件の整理、およびアプローチについて述べる。

3.1 EPC ネットワーク

本節では、本研究が前提とする EPC ネットワークについて述べる。EPC ネットワークとは、現在多くの製品に付いているバーコードの次世代を担う製品識別技術の開発を主目的として提案されたシステムである。

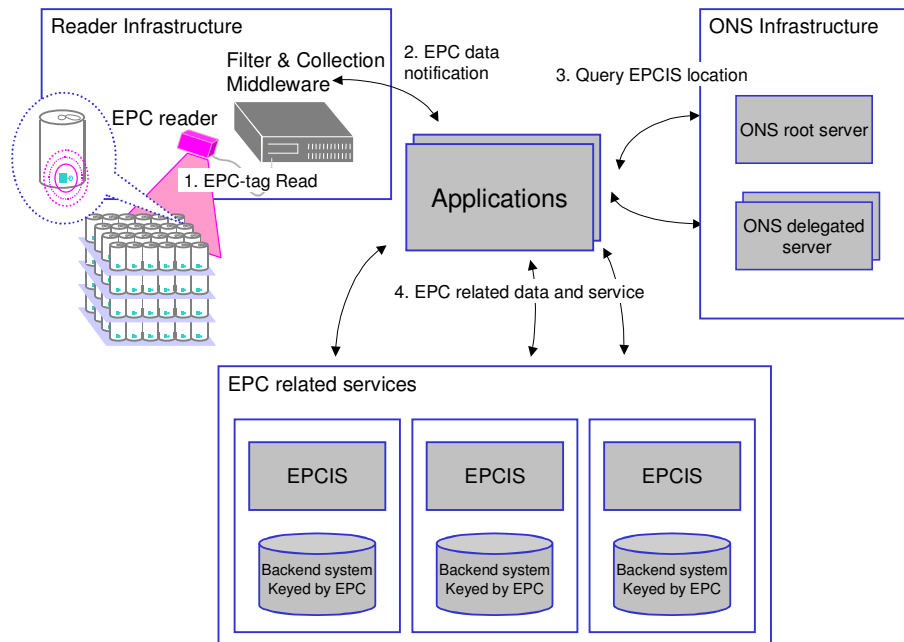


図 3.1: EPC ネットワーク概要図

EPC ネットワークは、図 3.1 のようなアーキテクチャ構造を持つ。EPC ネットワークを利用するアプリケーションを中心に、一意性が保証されている EPC を保持し製品類に貼付される EPC タグ、EPC タグの EPC を読み取るリーダ、リーダからの情報の集約とアプリケーションへの情報配送を行う Filter & Collection ミドルウェア群 [34]、EPC に基づく情報サービスを提供する EPCIS(EPC Information Service)[35]、アプリケーションに対して適切

な EPCIS へのマッピングを提供する ONS(Object Name Service)[36] などから構成される。

EPC ネットワークにおけるシステムアーキテクチャの各コンポーネント仕様については、現在も標準化に向けて議論が進められている。本研究では、EPC ネットワークの仕様そのものについては研究対象外であるため特に触れないが、本研究を進める上で特に重要となる以下の項目について述べる。

- 個体識別子の割り当て、管理
個体識別子の割り当て、管理とは、実際に様々な製品に ID を割り当ててにあたり、ID 空間を誰が (どの組織が)、どのように管理するのかという問題である
- 製品メタ情報の管理
製品メタ情報の管理とは、ID に関連付けられた製品情報 (例えば、製品の名前や、製造場所など) の管理権限に関する問題である

以上の 2 点について詳述する。

個体識別子の割り当て、管理

EPC ネットワークでは EPC と呼ばれる識別子を用いる。表 3.1 に EPC のコード体系を示す。

表 3.1: EPC コード体系

	Header	General Manager Number	Object Class	Serial Number
EPC (GID-96)	8bit 0011 0101 (Binary Value)	28bit 268,435,455 (Max Decimal Value)	24bit 16,777,215 (Max Decimal Value)	36 bit 68,719,476,735 (Max Decimal Value)

表 3.1 に示す通り EPC ネットワークでは、オブジェクトレベルでの利用、およびインスタンスレベルでの利用を想定し、EPC 自体が、製造者 (General Manager Number)、プロダクト ID(Object Class)、シリアルナンバー (Serial Number) というように構造化されたコード体系になっている。現在、EPC ネットワークでは、各製造者ごとに ID 空間を管理することが想定されている。本研究では、オブジェクトを識別するにあたり一意性が保証された EPC を利用するが、EPC の割り当て・管理に関しては本研究の対象外とする。

製品メタ情報の管理

上述した通り、EPC ネットワークでは基本的に EPC の割り当て、管理は製品毎に各製造者が行う。したがって、各製品のメタ情報、すなわち製品名や、製造年月日、製造場所といった情報は各製造業者が管理することが現在考えられている。そして、トレーサビリティの実現にあたっては、各企業間の信頼を前提として、信頼しあう企業間での情報流通が想定されているが、データ管理用のデータフォーマットやデータ交換プロトコルなどについては標準

化の最中である。よって、本研究ではオブジェクトについての EPC をもとに製品のメタ情報を取得できることを想定しているが、データ管理権限や、データ取得プロトコルについては研究の対象外とする。

3.2 RFID の利用モデル

本節では、最初に RFID システムについて述べた後、本研究において想定している環境について述べる。

3.2.1 データキャリア型 RFID システムとネットワーク型 RFID システム

RFID を用いたシステムは、物に関係する情報の管理方法に注目すると、データキャリア型 RFID システムとネットワーク型 RFID システムの二つに分類できる。ここでは、この二つの RFID システムについて説明する。

- **データキャリア型 RFID システム**

データキャリア型 RFID システムでは、RFID タグ自体に ID 以外の情報やデータが書き込まれている。一般的にデータキャリア型 RFID システムで用いられる RFID タグは、タグ自体が保持できるデータ量が多くネットワーク型 RFID システムで用いられるタグよりも高機能であるという特徴が挙げられる。

物自体に情報を運ばせるという考え方には問題はないが、実際に利用する際には、その物が目の前になければ情報を処理できない、RFID タグのメモリ容量により情報量が制限される、書き込み時間が長く実運用が困難であるなどの難点を抱える。

- **ネットワーク型 RFID システム**

ネットワーク型 RFID システムでは、RFID タグ自体には ID 以外の情報は書き込まれない。ID 以外の情報やデータはネットワーク上に展開される。データキャリア型 RFID システムで用いられるタグと比較してタグが保持できるデータ量は少ないが、機能がシンプルであるためタグ単価が安く押えられるというメリットがある。コストの面やトレーサビリティの観点から、SCM の分野で現在バーコードの代わりとして期待されているのもネットワーク型 RFID システムである。

本研究では導入コストや、これまでの研究活動における運用実績なども考慮し、ネットワーク型 RFID システムを前提とする。

3.2.2 ネットワーク型 RFID システムの構成要素

ネットワーク型 RFID システムは、以下に挙げる要素から構成される。

- **RFID タグ**

全てのオブジェクトを識別するための ID を保持しているタグ。この RFID タグが様々なもの、建物や公共の場所などに貼付されたり、また人間がタグを携帯したりすることであらゆるオブジェクトを識別することが可能である。

- RFID リーダ

一意の識別子，すなわち RFID タグから ID を読みとるための RFID リーダは，無線などを通して直接，または何らかの通信機器を機器を経由して間接的にネットワークにつながっている．ネットワーク型 RFID システムにおいてリーダから取得される情報は，読み取った ID とその ID を検出したタイムスタンプの二つである．

また，RFID リーダは，リーダの所有という観点に注目すると，以下の二種類の運用方法が考えられる．

- － 個人所有型
携帯型の RFID リーダを個人が持ち歩くタイプ
- － 場所・環境設置型
部屋や棚，あるいは公共の場所などに RFID リーダが設置されているタイプ

- メタ情報データベース

前節で述べた通り，ネットワーク型 RFID システムでは，オブジェクトに関する情報は，ID 以外全てネットワーク上に展開される．よって，ネットワーク上に ID 以外のオブジェクトメタ情報を蓄積しておくためのメタ情報データベースが必要となる．

3.2.3 RFID 利用方法の検討

本節では，ネットワーク型 RFID システムを利用するにあたり，その利用形態について，利用コストやプライバシーの観点から考察を行う．利用形態について考えた場合，RFID タグとリーダの所属という観点から，以下の二つの利用形態が考えられる．

- ユーザ：RFID タグ，公共インフラ：RFID リーダ

人間が RFID タグを持ち，人間の周辺環境に RFID リーダがインフラとして設置されているモデル．このモデルでは，ネットワークとつながっている RFID リーダが環境側にあるため，リーダの管理者以外は特に何もする必要がないというメリットがユーザにある．

- ユーザ：RFID リーダ，公共インフラ：RFID タグ

人間が RFID リーダを持ち，人間の周辺環境に RFID タグが貼付されているモデル．このモデルでは，システムの基盤として多くのリーダを設置する必要がなく，環境に安価なタグを貼付するだけで良いため，設置コストが低いという利点がある．また，RFID システムを考える上でしばしば懸念されるプライバシーの観点から考えても，盗み見ることが可能な ID がないので有利である．

これらを踏まえ，二つの利用形態を比較したものが表 3.2 である．

以上の比較から，設置コストが低いこと，およびプライバシーの面で優位であることを重視し，本研究では後者の，人間が RFID リーダを所持して行動し，人間の周辺環境に RFID タグが多数存在する環境を想定する．

表 3.2: RFID 利用モデルの比較

	人がタグを持ち，環境内にリーダが付設	人がリーダを持ち，環境内にタグが遍在
設置コスト	高い	低い
新サービスの導入	エンドユーザが特にすべきことはない	クライアント側でサービスを受けるアプリをメンテナンス（インストールや設定）する必要あり
プライバシー	電波による ID の盗み見などが可能であり不利	電波による ID の盗み見などできない点で有利

3.3 複合コンテキストモデルの提案

1.3 節において，本研究におけるコンテキストという単語を“あるオブジェクトを識別することによって得られる，人間の行動に関して意味付けされた情報”と定義した．一般に，人間の行動は百者百様であるため，コンテキストアウェア分野では，計算機上で人間の行動をどう表現するかという部分が大きな課題となってきた．例えば，ある人が今向いている方向や，外出するといった行動をセンサを用いて認識するのに，既存のコンテキストアウェアシステムでは，様々なセンサデバイスを用いて時系列上のある一点におけるセンサの状態とコンテキストを結びつけるといった方法をとってきた．しかし，実空間上のオブジェクトを利用してコンテキストを取得することを考える場合，時系列上のある一点における状態のみからコンテキストを把握することは困難である．

よって，本研究では，第 2 章で整理した既存システムにおける問題を踏まえ，複合コンテキストモデルの提案を行う．

本研究における複合コンテキストを以下のように定義する．

複合コンテキスト = 複数の連続したイベントの集合から判断・生成されるコンテキスト情報

これは，図 3.2 に示す通り，センサ技術を高度化するのではなく，RFID などの個体識別情報を組み合わせてコンテキストを判断するモデルである．

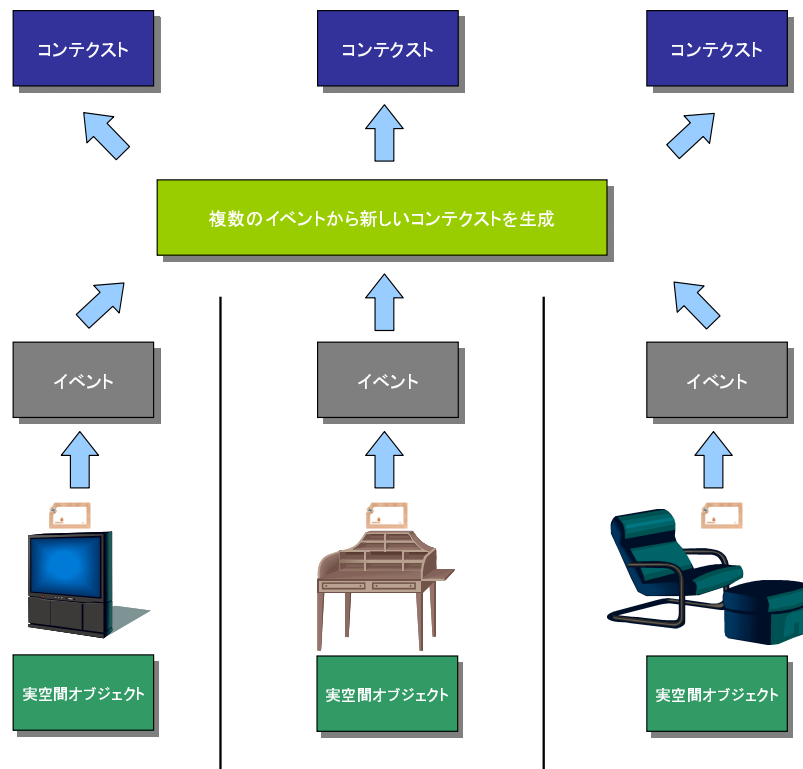


図 3.2: 本研究が目指すモデル

これまで述べてきたように、SCM 分野での RFID 導入の流れを背景として、様々なオブジェクトにタグが貼付されることが予想され、単純な個体識別のイベントは容易に取得できるようになると考えられる。この個体識別イベントを利用したコンテキストウェアシステムを考えた場合、センサ技術を高度化したり、多種多様のセンサを設置したりしない分コストは安いが、その反面、センサから取得される情報は限られてくる。よって、センサから取得されたイベント情報をもとに、システム側でコンテキストを判断する機構が必要となる。

そこで、本節ではオブジェクトと人間の行動の関係について述べた後、オブジェクトからコンテキストを生成するにあたり、考慮すべき点について整理する。

3.3.1 オブジェクトと行動

人間が何か行動を起こす時、人間は様々なオブジェクトに触れることが多いと考えられる。例えば、何かを“食べる”という行動を考えた場合、箸や茶碗といったオブジェクトに必ず触れるであろうし、“電車に乗る”という行動であれば、切符や定期券に必ず触れるであろう。ある行動を起こす時に、“オブジェクトに触れること”を RFID を用いて“オブジェクトを検出すること”と置き換えることは可能である。ゆえに、RFID を利用することによって、実空間上のオブジェクトから人間の行動意図、すなわちコンテキスト情報を取得することは可能であると考えられる。

このように、本研究では RFID を用いてオブジェクトから得られるコンテキスト情報を利用して人間の行動意図を推測するが、全ての行動をカバーできるわけではない。2.1 節で述べたように、例えば、ある人が現在どちらの方向を見ているかという情報や、ある空間内の

どこにいるかという情報の取得は困難である。

3.3.2 オブジェクトを利用したコンテキスト生成・表現に関する考察

実空間オブジェクトを利用してコンテキストを生成し、表現する際に考慮すべき問題として、以下の 3 点が挙げられる。

- 時間連続性
人間の行動において、ものを手に取る一連の流れに起因する問題
- オブジェクトの識別粒度
あるオブジェクトを認識する際に人間が意識する抽象度の違いに起因する問題
- 行動パターンのルールセット
コンテキストを生成するにあたって必要となる行動に関するルールセット表現に起因する問題

各項目について以下に詳述する。

時間連続性

2.1 節で述べた通り、オブジェクトを利用して生成するコンテキスト情報は、個人毎に異なるものであり、またその中でもイベントの検出時間帯などの要素によって違ったコンテキストとなり得る。複数のオブジェクト検出イベントからコンテキストが生成される例を考えてみる。例えば、家や車の鍵と携帯電話がある一定時間の間に検出され、その検出イベントの集合が“外出”であると登録されていたとする。この場合、鍵が先に検出されても、携帯電話が先に検出されても、その検出順序によって“外出”というコンテキストが変化することはないであろう。本研究では、このように検出順序によって影響を受けない、あるオブジェクト検出イベント群の集合を“組み合わせ”と定義する。我々の生活環境を考えた場合、このようなイベントの“組み合わせ”で表現できるコンテキストは非常に多いと考えられる [37]。

しかし、ある部屋 A のドア a と、ある部屋 B のドア b、そしてある部屋 C のドア c という 3 つのオブジェクト検出イベントからコンテキストを生成する例を時間軸に沿って考えてみると、“a が検出された後に、b が検出される”という検出パターンと“b が検出された後に、a が検出される”という検出パターンでは明らかにコンテキストが異なる。前者は“部屋 A から部屋 B へ移動した”と解釈でき、後者は“部屋 B から部屋 A へ移動した”と解釈できる。このように、時間軸を考慮して複数のイベントからコンテキストを生成することを考えた場合、同じイベントの組み合わせでも、異なったコンテキストが生成される可能性がある。本研究では、検出順序によってコンテキストが大きく変化すると考えられるために、時間軸に沿って厳密にその順序が表現される必要があるイベント群の集合を“順列”と定義する。

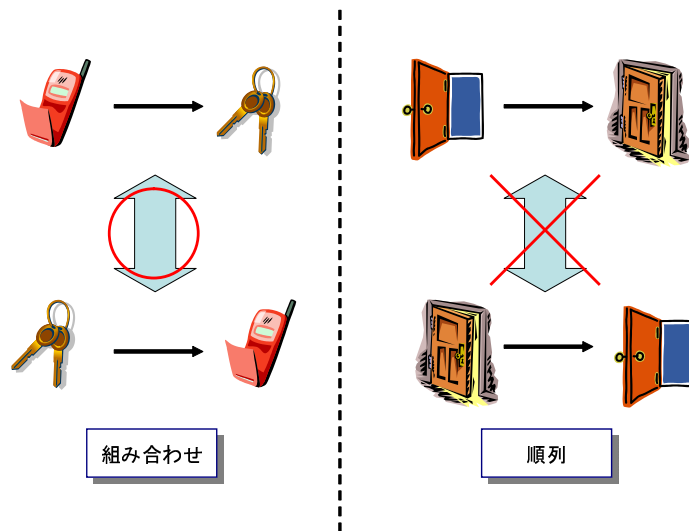


図 3.3: 時間連続性によるコンテキスト変化

オブジェクトの識別粒度

あるオブジェクトについて着目した場合、オブジェクトのとらえ方によって2種類の識別方法をとることが可能である。すなわち、あるオブジェクトを見たときの一般的な見方である製品単位での識別と、その製品の中からある特定の一つの製品を指す個品単位での識別が考えられる。具体的に述べると、ある机があったときに、それを“研究室にあるA社製の机”というように製品単位で認識するか、“私が研究室で現在使っている個体番号12345678が付いた机”という一意な個品単位で認識するかということである。本研究では、前者をオブジェクトレベルでの識別、後者をインスタンスレベルでの識別と呼ぶこととする。

これを踏まえ、オブジェクトからのコンテキスト生成について考察する。例えば、ある人がパジャマを着た状態で、かつ歯を磨いたら、それは“就寝する”というコンテキストであると定義したとする。人によって寝るときに触れるオブジェクトは異なるだろうが、今の例では、パジャマが常に一意で同じものしかないということは考えにくい。多くの人にとって、パジャマAと、パジャマBと、パジャマCというように同じ種類の製品であっても、複数のものを同じ場面の中で使用することは一般的な事象である。そのような場合、インスタンスレベルごとにコンテキストを定義するのは、イベントの全組み合わせパターンを定義・管理する必要があり、システムとして効率が悪い。

また、日常生活における食品や歯ブラシなどの消耗品を考えた場合も、インスタンスレベルでコンテキストを登録しておく、買い替えなどした際にデータを登録しなおす必要が発生し、効率が悪い。

よって、システムは、オブジェクトレベルの識別と、インスタンスレベルの識別の両方を扱える必要がある。

行動パターンのルールセット

オブジェクトの検出イベントを利用して、システムがコンテキストを判断するためには、オブジェクトの検出パターンを何らかの形でシステムが保持している必要がある。

人間の行動を連続する時間軸に沿って見たときに、オブジェクトの“組み合わせ”と“順列”という 2 つのパターンがあると先程述べた。しかし、この 2 つのパターンを表現するだけでは、システムとして十分ではない。

図 3.4 を例に考えてみる。図 3.4 は筆者が研究室にいる時の日常的な行動パターンの一部を表している。

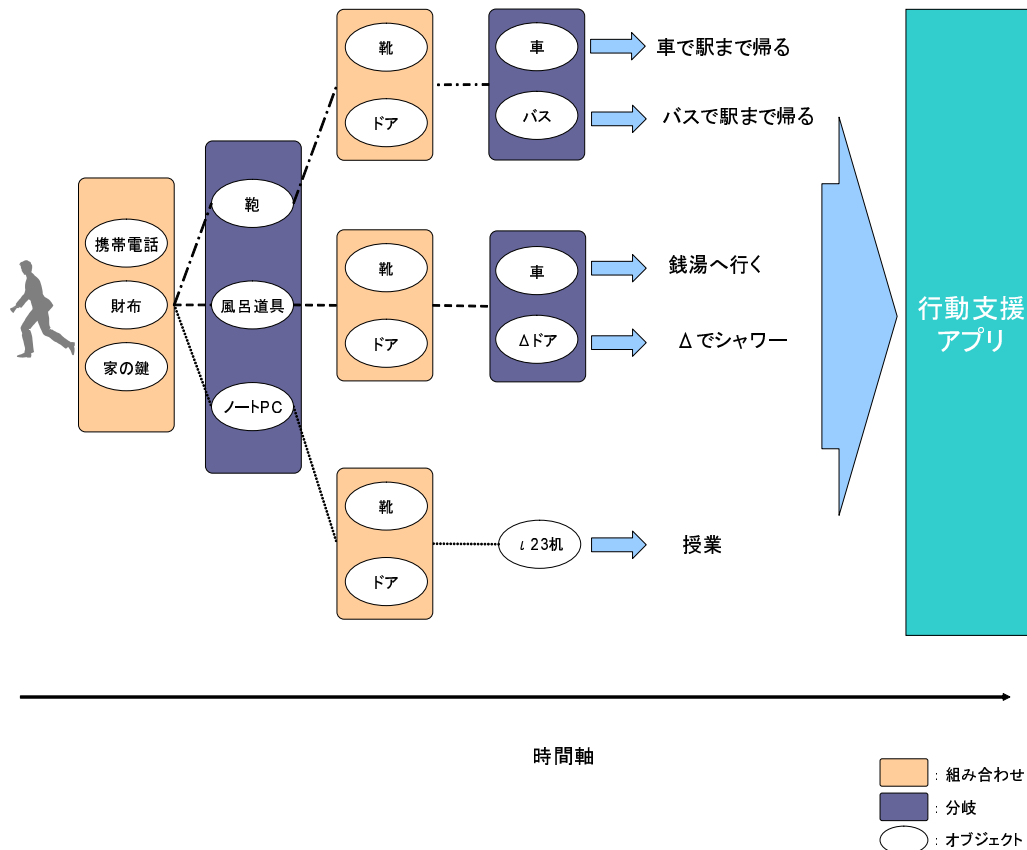


図 3.4: イベントによってコンテキストが分岐する例

図 3.4 では、オブジェクトの検出イベントの並びから、コンテキストが 5 つに分岐している。この例を、組み合わせだけで表現するには全てのパターンを記述する必要がある。人間のまわりにあるオブジェクトは数えきれない程存在するため、全てのイベントパターンを個別に保持するのは現実的ではない。図 3.4 に示したように、ある別々のコンテキストであっても、それを構成するイベント群を見たときに重複する部分が存在することは多々あると考えられ、これらを統合して表現するルールセットの記述を検討する必要がある。

3.4 機能要件の整理

本節では、前節で述べた複合コンテキストモデルを実現するにあたり、必要となる機能要件について整理する。

オブジェクト固有のコンテキスト表現

前節で述べたように、オブジェクトを利用した複合コンテキストの利用にあたって、時間軸に沿った複数のイベントからコンテキストに変換できること、およびオブジェクトの粒度の違いに対応したコンテキスト表現ができる必要がある。

時間連続性を考慮したコンテキスト表現

まず、時間連続性の問題に対するアプローチを述べる。人間が行動を起こすにあたって、オブジェクトを検出する。この検出パターンについて、実時間性の観点から見た時に、図 3.5 に示すように“ 順列 ”と“ 組み合わせ ”、“ 分岐 ”で表現できると考えられる。

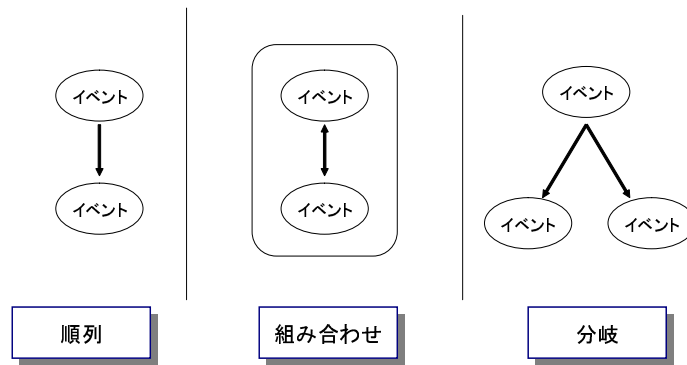


図 3.5: イベントのルールセット

そして、これらのルールセットを組み合わせることで、人間の行動を疑似的にツリー構造に見立てることができる。すなわち、ツリーの階層が上から下へというベクトルによって順列を表現し、組み合わせはツリーのある一階層に、分岐はツリーの枝分かれにそれぞれマッピングすることで表現する。

オブジェクトのメタ情報粒度を考慮したコンテキスト表現

前節において、インスタンスレベルのみでコンテキストを定義することの問題について述べた。この問題を解決するために、オブジェクトレベルとインスタンスレベル両方を用いてコンテキスト情報を定義できる必要がある。本研究では、EPC ネットワークのある環境を想定しており、センサイベント情報に含まれる EPC を検索キーとして、検出したオブジェクトの名前を取得し、データベースに登録することでオブジェクトレベルでの管理を実現する。

3.5 まとめ

本章では、まず最初に本研究の前提とする EPC ネットワークについて概説し、利用モデルを、データキャリア型 RFID システムとネットワーク型 RFID システム双方の観点から比較検討した。そして、実空間上のオブジェクトを識別することが、コンテキスト情報となり得ることを説明した。最後に、第 2 章で挙げた問題を解決するために、アプリケーション開

発の基盤となるミドルウェアの提案を行い、機能要件について整理を行った。

次章では、複数コンテキストを容易に扱うためのミドルウェアについて具体的な設計を行う。

第4章 複合コンテキスト管理ミドルウェアの設計

本章では，第3章で整理した機能要件に基づき，複合コンテキスト管理ミドルウェアの設計について述べる．

4.1 設計概要

本節では，複合コンテキスト管理ミドルウェアの設計概要について述べる．

第3章では，本研究で提案する複合コンテキストモデルの機能要件として，オブジェクト固有のコンテキスト表現，時間連続性を考慮したコンテキスト表現，オブジェクトのメタ情報粒度を考慮したコンテキスト表現が必要であると述べた．これらの機能要件から，複合コンテキストモデルを実現するためには，RFID リーダから送信されるイベント情報から特定のコンテキストを生成してアプリケーションへ提供する機能が必要である．この機能はリーダーとアプリケーションの間を仲介するという役割であり，ミドルウェアとして実装する手法が最適である．そこで本研究では，イベント情報からコンテキストを生成する複合コンテキスト管理ミドルウェアを提案する．

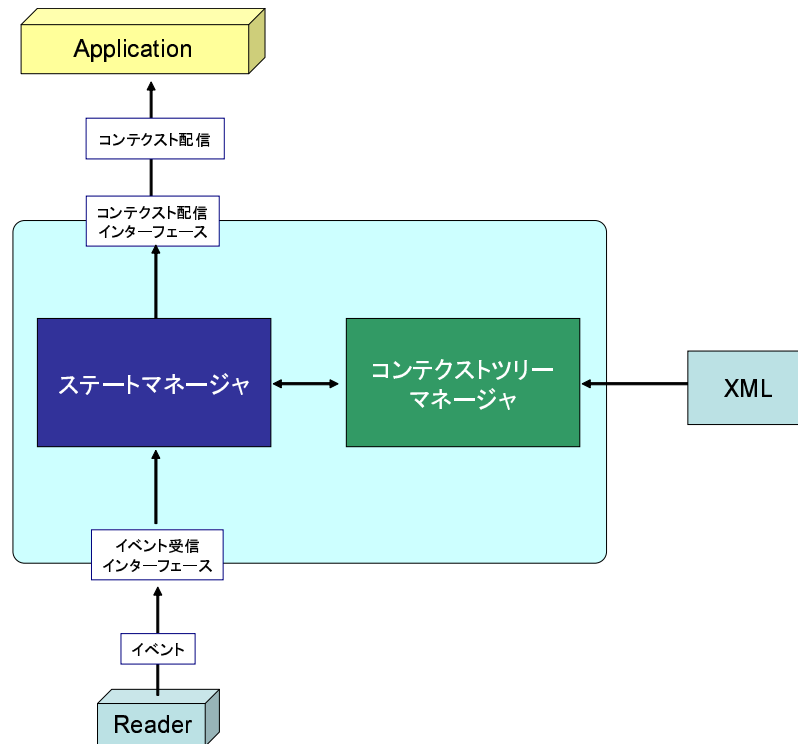


図 4.1: ミドルウェア構成図

図 4.1 は、複合コンテキスト管理ミドルウェアの全体を表す設計図である。本ミドルウェアが提供する機能を以下の 3 つのモジュールに分けて設計した。

- イベント受信インターフェース、コンテキスト配信インターフェース
- コンテキストツリーマネージャ
- ステートマネージャ

イベント受信インターフェースおよび、コンテキスト配信インターフェースは、ミドルウェアの持つ基本的な機能としてリーダとアプリケーションとの通信を行うためのインターフェースである。イベント受信インターフェースはリーダに提供されるインターフェースであり、リーダから送信されるイベント情報を受信する。コンテキスト配信インターフェースはアプリケーションに提供されるインターフェースであり、コンテキスト情報を必要とするアプリケーションからの接続要求に応答して、生成されたコンテキストを配信する。コンテキストツリーマネージャは、ユーザによって記述される XML ファイルを元にコンテキストを導出するためのツリーを生成・管理するモジュールである。ステートマネージャは、受信したイベント情報が一致するかどうかを調べて、ツリー内のどの節まで辿ったのかを記録するためのモジュールである。

以下に各モジュールの設計に関して詳述する。

4.2 ミドルウェアの基本機能

本節では、本研究で提案する複合コンテキスト管理ミドルウェアの基本機能に関する設計について述べる。

本ミドルウェアは、リーダとアプリケーションの仲介を行うために、リーダと通信を行うためのイベント受信インターフェース及び、アプリケーションと通信を行うためのコンテキスト配信モジュールが必要である。

4.2.1 イベント受信インターフェース

イベント受信インターフェースは、リーダから送信されるイベント情報を取得してステートマネージャへ渡す機能を提供する。

イベント情報を送信するリーダは前提条件として、ユーザが携帯することを想定している。また、本ミドルウェアはアプリケーションに対して安定した情報提供を行う必要があるために、ミドルウェアサーバ上で動作することを想定している。ユーザがリーダを携帯する場合は、リーダの物理的な位置が固定されないため、ネットワーク的な位置も常に移動する可能性がある。そのためリーダは無線 LAN など移動先にあるリソースを利用してネットワーク経由で情報を送信すると考えられる。そこで、イベント受信インターフェースはネットワークから情報を取得するためにソケットを利用する。

また、ユーザが携帯するリーダの数が複数である可能性が考えられる。リーダが複数存在する場合には、それぞれ別のリーダ制御プログラムが動作して、ミドルウェアサーバに対して複数の接続要求を行う。このため、本ミドルウェアはマルチクライアントに対応させ、複数のリーダから送信されるイベント情報を処理可能にする。

4.2.2 コンテキスト配信インターフェース

コンテキスト配信インターフェースは、アプリケーションにコンテキスト情報を配信する機能を提供する。

本モデルではアプリケーションの動作環境については規定していない。これは、アプリケーションがミドルウェアと同一ホスト上で動作しているのか、ネットワーク的に離れたホスト上で動作しているのかわからないということである。そこでコンテキスト配信インターフェースはイベント受信インターフェースと同様に、どちらの場合でも通信が行えるようにソケットを利用する。

また、コンテキストを利用するアプリケーションが複数動作している可能性が考えられる。複数のアプリケーションが動作している場合には、接続要求をしてきた全てのアプリケーションに対してコンテキストを配信する必要がある。そこでコンテキスト配信インターフェースでは、接続してきたアプリケーションを全て記憶しておき、コンテキストが生成された場合には全てのアプリケーションに配信できるようにする。

4.3 コンテキストツリーマネージャの設計

本節では、コンテキストツリーマネージャに関する設計について述べる。

コンテキストツリーマネージャは第 3 章で述べた機能要件内の時間連続性を考慮したコンテキスト表現を実現するためのモジュールである。

4.3.1 コンテキスト生成の条件

3.4 節で述べたように、時間連続性を考慮したコンテキストを表現するためには“組み合わせ”、“順列”、“分岐”という 3 つの抽象的概念を表現する必要がある。

順列は、イベント情報に順番を振ることによって表現することが可能である。例えば、イベントの前後関係を『→』で区切ることによって表現する場合には、図 4.2 で示すように表すことができる。

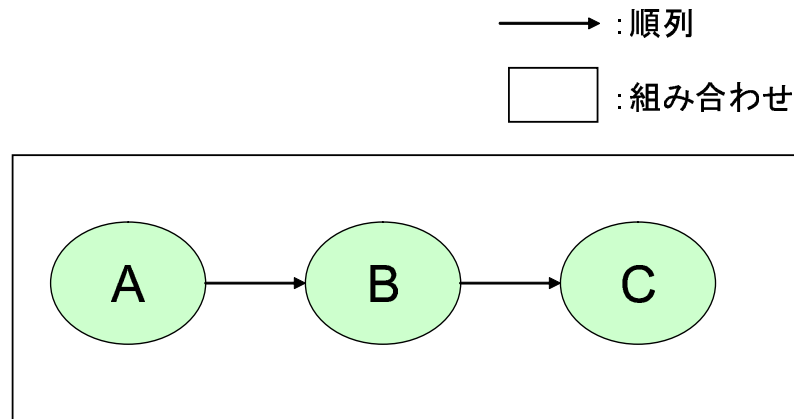


図 4.2: 順列

組み合わせは、順番は必要ないが複数のイベントがグループになっていることを表現する必要がある。例えば、イベントのグループを『□』で囲むことによって表現する場合には、図 4.3 で示すように表すことができる。

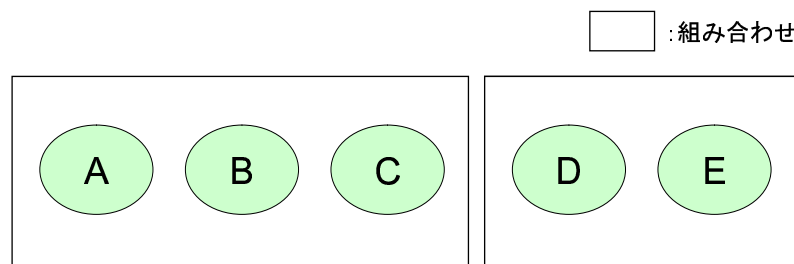


図 4.3: 組み合わせ

さらに、順列と組み合わせは図 3.4 で示したように混在する。図 4.4 に順列と組み合わせが混在する場合の図を示す。

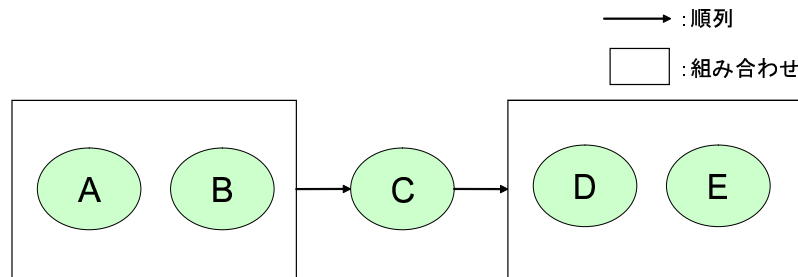


図 4.4: 順列と組み合わせ

本ミドルウェアでは、以上のようなコンテキストを生成するための条件を保持しなければならない。

4.3.2 ツリー構造

本ミドルウェアでは、コンテキスト生成の条件をツリー構造で保持する。

最も簡単なコンテキスト生成の条件を保持する方法は、リストで保持する方法である。もしコンテキスト生成の条件をリストで保持する場合には、全てのイベントパターンを設定ファイルに記述しておき、プログラム内部ではリンクドリストとしてメモリにマップする。そしてイベント情報を受信した場合には、全てのイベントパターンにおいてリニアサーチを行って、コンテキストの生成を判断しなければならない。

しかしこの方法は非常に効率が悪いため、条件が増えるにつれて劇的にパフォーマンスが低下する。またイベントパターンが複雑になり、設定ファイルの記述が困難であると考えられる。そこで本ミドルウェアでは、コンテキスト生成の条件をツリーで保持することによってこれらの問題点を解決する。ツリーによって条件を保持することの利点は、3.4 節で述べた“分岐”の概念が導入できる点である。この分岐を利用することによって、同じ条件が存在する場合には 1 つにまとめることができる。それにより、コンテキストの生成はツリーを辿っていくことによって実現できる。本研究では、このツリーのことをコンテキストツリーと呼ぶ。

4.3.3 XML による設定ファイルの記述

本ミドルウェアでは、コンテキスト生成の条件を XML によって記述する。

コンテキスト生成の条件はユーザによって設定ファイルに記述されることを想定している。そして本ミドルウェアはこの設定ファイルに基づいてコンテキストを生成する。設定ファイルには、前述した“組み合わせ”、“順序”、“分岐”を記述できることが条件となる。またプログラム内部ではツリー構造を利用するため、ツリー構造を表現できる記述手法が効率的である。これらの理由から設定ファイルは XML によって記述する。

4.4 ステートマネージャの設計

本節では、ステートマネージャの設計について述べる。

4.4.1 ステート

ステートとは、コンテキストツリーをどこまで辿ったのかを表す情報である。

本ミドルウェアでは、複数のイベントからコンテキストを生成するため、必要なイベントは保持しておかなければならない。しかし、リーダから送信されたイベント情報を時系列順に保持しておく方法では、全てのイベントパターンに関して一致しているかどうかを調べなければならない。例えば、4階層の条件だった場合には直前4つのイベント情報を取り出して、それぞれのイベントが一致しているかどうかを調べる必要があるため効率が悪い。

そこで、本ミドルウェアではステートを利用してツリーのどこまで条件が一致したかを保持する。図 4.5 で示すように、ステートはコンテキストツリーのどの節にいるのかを表すポインタである。イベント情報を受信すると、現在指し示している節の条件と一致しているかを調べ、もし一致していた場合は次の節へ移動し、一致していない場合にはコンテキストの条件不成立と判断して破棄される。コンテキストツリーの最下端である葉まで到達した場合には、そこに記述されているコンテキストが生成される。

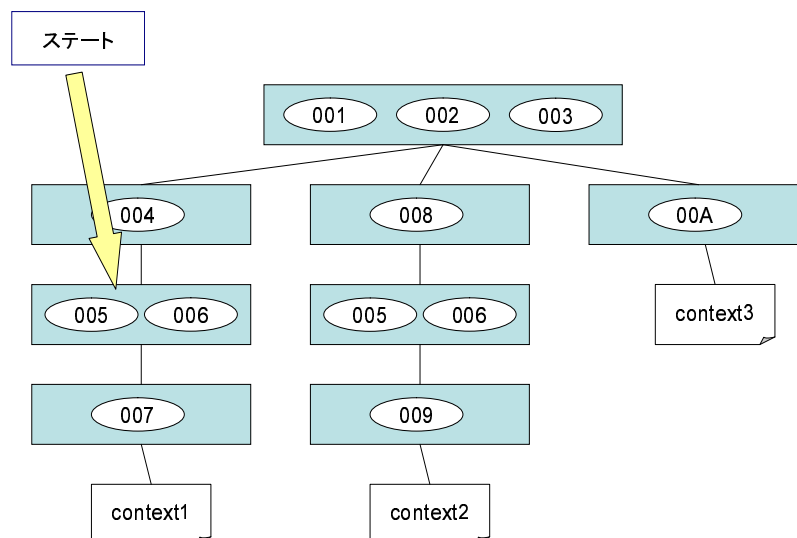


図 4.5: ステート

4.4.2 ステートリスト

本ミドルウェアでは、ステートがリストで保持される。

先に生成されたステートが最下端に到達するよりも前に、別の条件に一致するイベント情報が到達する可能性が考えられる。そのためステートは複数保持することが可能でなければならない。

また、ステートはイベント情報を受信する度に次の節に移動するための条件を満たしているかどうかを判断する必要があるため頻繁にアクセスされる。さらに全てのステートに関してチェックをしなければならない。そのため、ステートはプログラム内部でリストとして保持されている。

4.5 まとめ

本章では，複合コンテキスト管理ミドルウェアの設計について述べた．

次章では本章での設計を踏まえ，具体的な実装について述べていく．

第5章 複合コンテキスト管理ミドルウェアの実装

本章では，第4章で述べたシステム設計をもとに，複合コンテキスト管理ミドルウェアの実装について述べる．

5.1 システム構成

本節では，本研究で実装したミドルウェアのシステム構成について述べる．図5.1に本システムの構成を示す．

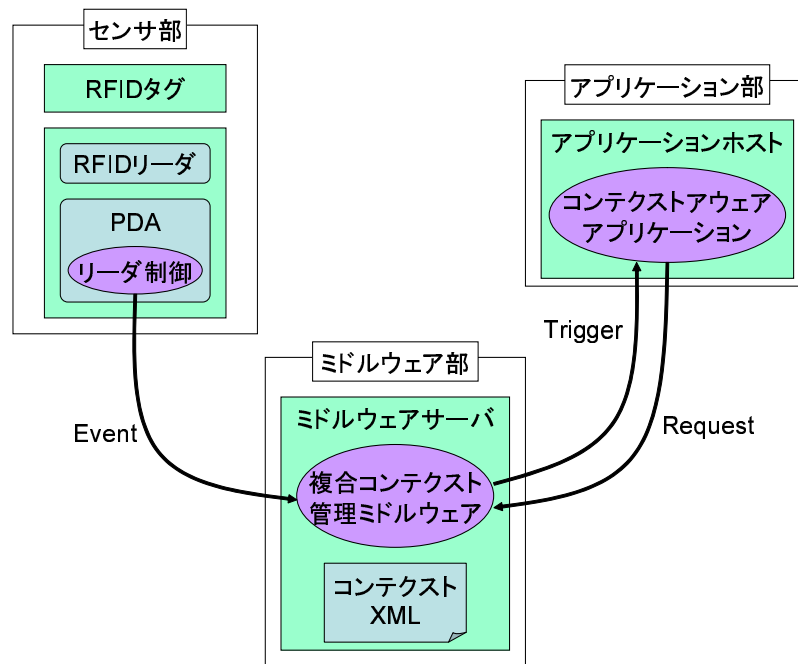


図 5.1: システム構成

本システムは，センサ部，ミドルウェア部，アプリケーション部によって構成される．本章の残りの節では，センサ部，およびミドルウェア部の実装について詳述する．

5.2 センサ部

センサ部は，RFID タグと RFID リーダから構成されている．第3.2節で述べた通り，本システムではネットワーク型 RFID システムを前提としており，今回は図5.2に示した omron

社の 13.56MHz 帯近接型 RFID タグを使用した。

本研究における想定環境において，RFID リーダは，タグの読み取り機能の他に以下が必要である。

- 携帯可能であること
- 無線 LAN による通信が可能であること
- 読み取った ID を規定のフォーマットで，ネットワークを経由してサーバへ送信できること

そこで，今回は，図 5.3 に示したように，CF 型 RFID リーダ，および CF スロットを搭載した PDA を利用して携帯型 RFID リーダを実装した．表 5.1 に CF 型 RFID リーダの仕様を，表 5.2 に PDA の仕様，およびリーダ制御プログラムの実装言語を示す。

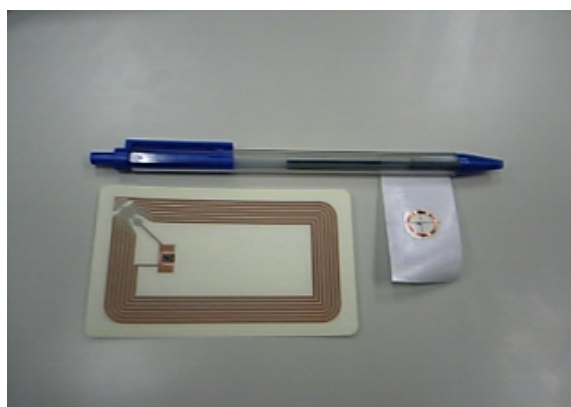


図 5.2: RFID タグ

表 5.1: CF カード型 RFID ユニット V720S-HMF01 仕様

発信周波数	13.56MHz
交信距離	3cm
上位インタフェース	コンパクトフラッシュ Type II (9600bps)



図 5.3: 携帯型 RFID リーダ端末

表 5.2: Fujitsu Pocket Loox FLX3AW 仕様, および実装言語

OS	Windows Mobile2003 for Pocket PC
CPU	Intel PXA255 400MHz
メモリ	RAM 128MB/ROM 64MB
通信機能	ワイヤレス LAN IEEE802.11b, Bluetooth
実装言語	eMbedded Visual C++ 4.0

PDA 上では, リーダ制御プログラムが動作しており, 検出された ID は, このプログラムによって本ミドルウェアに適合するフォーマットへ変換され, ミドルウェアサーバへ送信される.

本実装では, 図 5.4 に示した構造体をサーバへ送信する. 本研究におけるイベントは, PDA から送信される識別子とタイムスタンプで構成される. 本実装では, EPC ネットワークを前提としているため, 識別子として EPC を取得する.

```
struct event{
    char epc[24];
    u_time time;
};
```

図 5.4: リーダから送られてくるイベントの構造体

5.3 ミドルウェア部

表 5.3 に本研究で提案したミドルウェアの実装環境を示す。

表 5.3: 実装環境

OS	FreeBSD 6.0-Release
CPU	Pentium4 2.50GHz
Memory	1024MB
実装言語	C 言語

本実装におけるミドルウェア部は、コンテキストツリーマネージャ部とステートマネージャ部の 2 つから構成される。

- コンテキストツリーマネージャ部
複数のイベント群とコンテキストとのマッピングを保持する
- ステートマネージャ部
コンテキストツリーマネージャに対して、検出されたイベントの検索問い合わせを行う。

コンテキストツリーマネージャ部

コンテキストツリーマネージャ部では、libxml ライブラリ version2.6.22 を用いて、XML で記述されたコンテキスト設定ファイルを読み込む。これにより、イベントとコンテキストの対応付けを表すツリー構造でミドルウェア内に保持される。図 5.5 にコンテキスト設定ファイルを読み込むプログラムを示す。なお、図 5.5 について、スペースの関係で省略しているが、本システム中のコンテキスト設定ファイルは、正しい XML の文法に沿って記述されている必要がある。

また、図 5.6 に示したコンテキストツリーを例として、図 5.7 にコンテキスト設定ファイルの記述例を示す。

```

int load_xml(char *xml)
{
    if ( xml == NULL )
        return -1;

    doc = xmlReadFile(xml, NULL, 0); //xml ファイルの読み込み
    if ( doc == NULL )
    {
        fprintf(stderr, "ERR: could not parse file %s\n", xml);
        return -1;
    }

    root = xmlDocGetRootElement(doc); //root エLEMENTの取得
    if ( root == NULL )
    {
        fprintf(stderr, "ERR: could not get root element %s\n", xml);
        return -1;
    }

    return 0;
}

```

図 5.5: XML ファイルパーサ関数

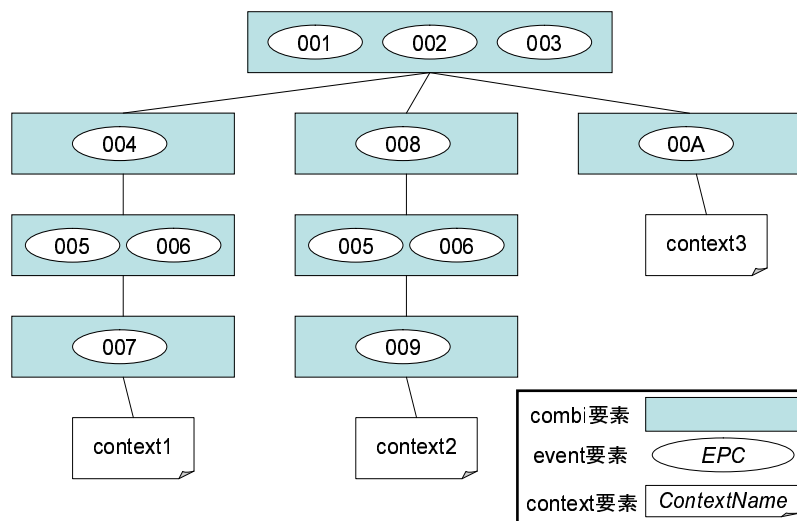


図 5.6: コンテキストツリー例

```
<?xml version="1.0" encoding="shift_jis"?>

<combi>
  <event>
    <epc>1234567812345678000000001</epc>
    <name>携帯電話 </name>
  </event>
  <event>
    <epc>1234567812345678000000002</epc>
    <name>財布</name>
  </event>
  <event>
    <epc>1234567812345678000000003</epc>
    <name>鍵</name>
  </event>

  <combi>
    <event>
      <epc>1234567812345678000000004</epc>
      <name>靴</name>
    </event>
    <combi>
      <event>
        <epc>1234567812345678000000005</epc>
        <name>Z202 ドア</name>
      </event>
      <event>
        <epc>1234567812345678000000006</epc>
        <name>靴</name>
      </event>

      <combi>
        <event>
          <epc>1234567812345678000000007</epc>
          <name>レガシィ</name>
        </event>
        <context>context1</context>
      </combi>
    </combi>
  </combi>
<!-- 以下省略 -->
```

図 5.7: コンテキスト設定ファイル例

ステートマネージャ部

ステートマネージャ部は、センサ部から受信したイベントを、コンテキストツリーマネージャ部に問い合わせ、イベントパターンが完全に一致したと判断された場合、アプリケーションに対してトリガを送信する。今回のプロトタイプ実装では、コンテキストツリーの最下層にコンテキスト情報を配置した。

ステートマネージャ部では、センサ部からの接続を受け付けると同時に、アプリケーションからの接続も受け付ける。センサ部から受信したイベントを、コンテキストツリーマネージャ部に問い合わせ、ステートの保持・更新を行う。イベントパターンが完全に一致したと判断された場合、アプリケーションに対してトリガを送信する。アプリケーションに送信するトリガは、どのコンテキストが該当したのかについて、アプリケーション側で判別できれば良いため、既存システムでは文字列情報をアプリケーションに送信しているものが多い。本実装では、context1,context2 というように決まった形式の文字列をアプリケーションに渡し、その利用方法についてはアプリケーションが判断することとした。

今回のプロトタイプ実装では、図 5.8 に示したステート管理構造体を用いて、ステート管理を行った。

```
struct context_tree_pointer
{
    xmlNodePtr cur_node;
    int nevent;
    unsigned char *event_flag;
    unsigned char clear_flag;
    struct context_tree_pointer *next;
};
```

図 5.8: ステート管理構造体

ステート管理構造体を構成する各変数について述べる。

- cur_node
cur_node は、現在ステートによって管理されている節を指す。
- nevent
nevent は、ステート管理しているツリー構造の節にあるイベントの数である。
- event_flag
event_flag は、あるイベントの検出が通知された時点でステート管理しているツリー構造の節にあるイベントの一つに対するステートを表現する。
- clear_flag
clear_flag は、あるイベントの検出が通知された時点でステート管理しているツリー構

造の節にあるイベントがすべて検出されたか表現する。すなわち、ある時点で管理している階層に一つのイベントしかなければフラグが TRUE になり、ステートは次の階層を管理する。しかし、管理している階層が“ 組み合わせ ”である場合、組み合わせを構成するイベントが全て検出されない限り、次の階層のステートを管理することはない。

- next

next は、現在保持されているステートをリスト構造で表すためのポインタである。

ステートマネージャ部が、コンテキストツリーマネージャ部に対して、イベントの検索を行う際に使用する関数の一部を図 5.9 に示す。

```
int search_list(char *tid)
{
    .....

    /*
     * ツリー構造の最後までマッチしたらコンテキストをアプリケーションに送信
     */
    if ( is_clear(node->event_flag, node->nevent) == 0 )
    {
        clearbit_up(&node->clear_flag);
        if ( (context = search_context(node->cur_node->children)) != NULL )
        {
            send_context(context);
            tmp = *node;
            removeat(i);
            node = &tmp;
            i--;
        }
    }

    .....
}
```

図 5.9: コンテキスト検索関数

図 5.10 に、ミドルウェア部のフローチャートを示す。

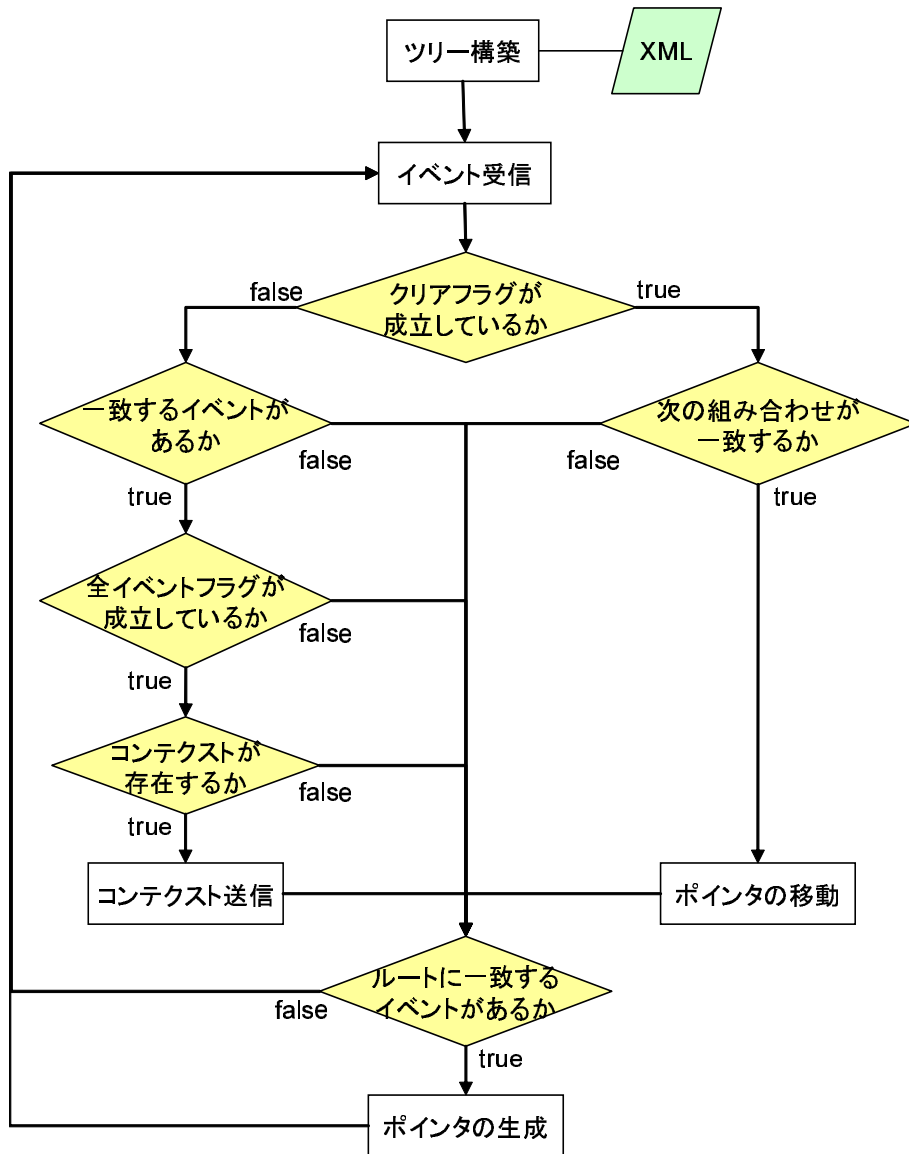


図 5.10: フローチャート図

5.4 まとめ

本章では，複合コンテキスト管理ミドルウェアを利用したシステムについて，センサ部，ミドルウェア部の実装についてそれぞれ述べた．

次章では，本システムの動作検証，および機能評価を行う．

第6章 評価

本章では、本研究で実装したシステムについて動作検証、および評価について述べる。

6.1 評価

本章では、まず最初に、本研究において実装したミドルウェアについて動作検証を行い、想定した環境においてシステムが問題なく動作することを確認する。そして、実装したミドルウェアが、本研究で想定した環境において機能要件を満たしているか検証し、想定した環境における本機構の有効性を示す。

6.2 動作検証

本節では、実装したミドルウェアが正しく動作するかについて検証した結果を述べる。具体的には、図 3.4 に示した筆者の日常を例として、本システムの動作検証を行った。付録 A に載せた定義ファイルを用いて、検証実験を行った。

ミドルウェアが動作している間、常にステートマネージャがステートの管理を行っている。その様子を図 6.1 に示した。図 6.1 では、イベントが検出される毎に、ステートマネージャはコンテキストツリー部に問い合わせを行い、ステートの更新を行っている。そして、コンテキストツリーの最下層に到達すると、検出したイベント群があるコンテキストを表す動作だと判断し、アプリケーションにトリガを送信できた。また、ステートで管理しているイベント以外のイベント問い合わせが発生した時、ステートを更新できることも確認した。

以上の検証より、本ミドルウェアは想定したとおりの動作を行えることが確認できた。

6.3 機能比較

本節では、実装したミドルウェアの機能について評価した結果を述べる。

3.4 節において、本研究が対象とする環境では、以下の 2 つが必要であると述べた。

1. 時間連続性を考慮したコンテキスト表現
2. オブジェクトのメタ情報粒度を考慮したコンテキスト表現

実装したミドルウェアでは、時間軸に沿ったイベントの並びをツリー構造に疑似的に見立て、データを保持することでコンテキストの判断が行えることができた。また、本実装では、コンテキストの設定を行うにあたり XML を用いた。これにより、イベントに拡張性を持たせることができ、オブジェクトのメタ情報粒度を考慮したコンテキスト表現を行うことが可能である。よって、本システムは機能要件を満たしている。

次に、コンテキストウェア分野における既存ミドルウェアとの比較を行う。表 6.1 に比較した結果を示す。

表 6.1: 既存ミドルウェアと本システムとの比較

	TEA	ANTH	Bazzar	本機構
複数イベントの考慮	○	△	×	○
実空間オブジェクトへの透過性	△	×	△	○
時系列概念	△	△	△	○

表 6.1 の各項目について述べる。

- 複数イベントの考慮
本ミドルウェアは、オブジェクトの特性について考察し、複数のイベントからコンテキストの判断が可能である。
- 実空間オブジェクトへの透過性
本ミドルウェアは、実空間オブジェクトを扱うことを念頭に置き、オブジェクト識別イベントの特性を考慮している。
- 時系列概念
本ミドルウェアは、複数のイベントを扱うにあたり、実時間性を考慮し、実際に扱える実装になっている。

6.4 まとめ

本章では、本研究で実装したミドルウェアについて、最初に動作検証を行い、意図した通りにシステムが動くことを示した。そして、本研究が想定する環境において、機能要件を満たしていることを示した。

```
./contextd -r 30001 -a 30002 -x context.xml
--- options ---
rport: 30001      ← リーダからイベントを受け取るポート
aport: 30002      ← アプリケーションに対してサービスを行うポート
xml: context.xml  ← イベントとコンテキストの定義を行う xml ファイル

----- 中略 -----

--- list ---
nevent: 3 clear: 0
100
nevent: 3 clear: 0
101
--- list ---
nevent: 3 clear: 0
010
nevent: 3 clear: 0
110
nevent: 3 clear: 1
111
--- list ---
nevent: 1 clear: 1
1
--- list ---
nevent: 2 clear: 0
10
--- list ---
nevent: 2 clear: 1
11
--- list ---
nevent: 1 clear: 1
1
Send [ context2 ]
state cleared...
--- list ---

----- 中略 -----
```

図 6.1: コンテキスト検索

```
%cat event.log
```

```
--- 中略 ---
```

```
800007B80000FA00000000001,1136887680  
800007B80000FA00000000003,1136887710  
800007B80000FA00000000002,1136887729  
800007B80000FA00000000004,1136887803  
800007B80000FA00000000006,1136887821  
800007B80000FA00000000005,1136887877  
800007B80000FA00000000007,1136888329  
800007B80000FA0000000000B,1136888875
```

```
--- 中略 ---
```

図 6.2: 検出イベントログ

第7章 結論

本章では、本研究のまとめと、今後の課題についての展望を示す。

7.1 まとめ

本研究では、実空間オブジェクトの検出イベント情報を複数利用して生成される、人間の行動に関するコンテキスト情報を複合コンテキストと定義し、この複合コンテキストを容易に利用するためのミドルウェアの設計と実装を行った。

これまで、コンテキストウェア分野において、コンテキストは、多種多様なセンサデバイスを環境側に設置し、人間の動きや位置情報といった人間の空間的な配置・状態に依存した情報で表されることが多かった。しかし、人間が行動の中で利用、あるいは触れることの多い実空間上のオブジェクトについて注目して考えると、同じ場所や動きの上であっても、そこに存在するコンテキストに大きな違いが発生し得る。

本研究では、今後普及が見込まれている自動個体識別技術を用いて、人間の行動に関するコンテキスト情報を取得することを提案した。そして、人間の行動に関するコンテキストをオブジェクトの検出イベントから生成するために、時間軸を考慮したイベントの順列、組み合わせ、分岐という3つの抽象化を行った。これらの検出イベントの並びを、ツリー構造で表現することを提案し、コンテキストをツリー構造で管理するミドルウェアを構築した。本システムを用いて、複数のイベントを複合的に扱うことで、人間の行動に関するコンテキスト情報を容易に利用することが可能となる。

7.2 今後の課題

本節では、今後の課題について展望する。

行動表現の多様性

本研究では、オブジェクトを用いて人間の行動コンテキストを表現するにあたり、順列、組み合わせ、分岐という3つの抽象化を行った。そして、複数のイベントデータをツリー構造で保持することでコンテキストを判断するミドルウェアのプロトタイプ実装を行った。

ミドルウェアにツリー構造でコンテキスト情報を保持させ、実験した結果、人間の行動コンテキストを表現できることが確認できた。しかし、人間の行動は、パターンにあてはまらない突然の行動、あるいは行動変化というものが常にあると考えられる。そのような事象について、今後実験を重ねることで、どの程度の行動であれば冗長性を持って対応できるのかについて検証していく必要がある。

アプリケーション開発

本研究では、複合コンテキストをミドルウェアで扱えるようにすることに主眼を置いたため、アプリケーションの開発には至っていない。しかし、アプリケーション開発を容易にしたことで、今後アプリケーションの開発が促進されることが期待される。

利用場面の拡大

本研究では、個人の行動支援を目的として、いくつか制約事項を加えることでシステムの有効性を示した。これらの制約事項を除くためには、どういった問題があり、どう解決できるかを考えていく必要がある。

謝辞

本研究を進めるにあたり、ご指導をいただきました慶應義塾大学 環境情報学部教授 村井純博士、および同学部助教授 中村修博士、同学部教授 國領二郎博士に感謝いたします。

また、本論文の執筆にあたり、多大なる御指導をいただきました慶應義塾大学 政策・メディア研究科講師 羽田久一博士、政策・メディア研究科助手 中根雅文氏、政策・メディア研究科講師 宇夫陽次郎博士、政策・メディア研究科助教授 三次仁博士、政策・メディア研究科研究員 鈴木茂哉氏、政策・メディア研究科助手 稲葉達也氏、環境情報学部専任講師 重近範行博士に感謝いたします。

学部から修士課程に至るまでの長きにわたり、日々の研究活動を通じて慶應義塾大学 政策・メディア研究科 博士課程 川喜田佑介氏には特にお世話になりました。改めてここに感謝の念を表したいと思います。

長い研究室生活において、常に苦楽を共にしてきた慶應義塾大学 政策・メディア研究科 修士課程 橋本和樹氏、成瀬大亮氏、山本聡氏、谷隆三郎氏に感謝いたします。彼らなくして私の修士生活はありえませんでした。本当にありがとう。

論文の追い込み時期を共にした慶應義塾大学 環境情報学部 芦阪浩輔氏、および本論文の執筆にあたりお手伝いいただきました慶應義塾大学 環境情報学部 佐藤泰介氏をはじめとする村井研究室の仲間に感謝いたします。

英語の面でサポートしていただきました谷岡洋平氏、山田奈穂氏、および学部卒業後も度々プレッシャーをかけていただきました高橋宏明氏、清水崇氏に感謝いたします。

最後に、これまで生活面において常に支えてくれた家族に感謝したいと思います。

本当にありがとうございました。

2006 年 1 月 12 日
廣瀬 峻

参考文献

- [1] M. Weiser. The computer of the twenty-first century. *Scientific American*, pages 94–104, September 1991.
- [2] インターネット ITS 協議会. <http://www.internetits.org/>.
- [3] Norman Adams Bill N. Schilit and Roy Want. Context-aware computing applications. *In Proceedings Workshop on Mobile Computing Systems and Applications*, pages 85–90, December 1994.
- [4] FeliCa. <http://www.sony.co.jp/Products/felica/>.
- [5] 電子マネー Edy. <http://www.edy.jp/>.
- [6] JR 東日本 Suica. <http://www.jreast.co.jp/suica/>.
- [7] JR 西日本 ICOCA. <http://www.jr-odekake.net/guide/icoca/>.
- [8] e-エアポート. <http://www.e-airport.jp>.
- [9] EPCglobal, 2003. <http://www.epcglobalinc.org>.
- [10] Auto-ID Labs, 2003. <http://www.autoidlabs.org>.
- [11] Anind K. Dey and Gregory D. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *HUMAN-COMPUTER INTERACTION*, Volume 16:97–166, 2001.
- [12] 青木 茂樹, 岩井 嘉男, 大西 正輝, 小島 篤博, 福永 邦雄. 人物の位置・姿勢に注目した行動パターンの認識. **電子情報通信学会技術報告**, pages 13–18, September 2002.
- [13] EPCGlobal. *EPC Tag Data Standard Version 1.1 rev 1.27*, May 2005.
- [14] 丸の内ユビキタスミュージアム. <http://tokyo-dmy.jp/p/>.
- [15] みまもりホットライン. <http://www.mimamori.net/>.
- [16] DoPa, NTT DoCoMo. <http://www.docomo.biz/dopa/>.
- [17] 角 康之, 江谷 為之, Sidney Fels, Nicolas Simonet, 小林 薫, 間瀬 健二. C-MAP: Context-aware な展示ガイドシステムの試作. **情報処理学会論文誌**, Vol.39, No.10, pages pp.2866–2878, October 1998.

- [18] R. Want, A. Hopper, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.
- [19] A. Hopper R. Want. Active Badge and Personal Interactive Computing Objects. *IEEE Transactions on Consumer Electronics*, February 1992.
- [20] A. Jones A. Ward and A. Hopper. A New Location Technique for the Active Office. *IEEE Personnel Communications*, 4(5):42–47, October 1997.
- [21] A. Chakraborty N. Priyantha and H. Balakrishnan. The Cricket Location-Support System. *MobiCom*, August 2000.
- [22] H. Balakrishnan N. Priyantha, A. Miu and S. Teller. The Cricket Compass for Context-Aware Applications. *MobiCom*, pages 1–14, July 2001.
- [23] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper and Mike Pinkerton. Cyberguide: A mobile context-aware tour guide. *Baltzer/ACM Wireless Networks*, 3(5):421–423, 1997.
- [24] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday and Christos Efstratiou. Experiences of developing and deploying a context-aware tourist guide: The guide project. *MobiCom*, pages 20–31, August 2000.
- [25] 森川 大補, 本庄 勝, 山口 明, 大橋 正良. コンテキストを活用したサービスプラットフォームの検討. *DICOMO2003*, June 2003.
- [26] 「タグ付けられた世界での行動を認識する携帯型アシスタント」の開発 -誰もが利用できるユビキタス基盤を目指して-, 2005. 独立行政法人情報処理推進機構 (IPA) 次世代ソフトウェア開発事業.
- [27] 楓 仁志, 山原裕之, 藤原聡子, 野口豊司, 東 辰輔, 島川博光. タグ付けられた世界における個人の行動特性を用いた意図推測. **組込みソフトウェアシンポジウム 2005**, October 2005.
- [28] 藤原 聡子, 島川博光. RFID アクセス履歴を用いた行動パターンのモデル化. **情報科学技術フォーラム (FIT2005)**, pages 35–36, September 2005.
- [29] Albrecht Schmidt, Kofi Asante Aidoo, Antti Takaluoma, Urpo Tuomela, Kristof Van Laerhoven, and Walter Van de Velde. Advanced interaction in context. *First International Symposium on Handheld and Ubiquitous Computing, HUC'99*, September 1999.
- [30] 鹿島 拓也, 猿渡 俊介, 森川 博之, 青山 友紀. 実空間プログラミングフレームワーク ANTH の設計と実装. **電子情報通信学会技術報告**, pages 447–452, March 2005.
- [31] Sun Microsystems, 2001. Jini Architecture Specification.
- [32] UPnP Forum. UPnP Device Architecture 1.0.

- [33] K. Fujinami, T.Nakajima. Towards System Software for Physical Space Applications. *Proceedings of ACM Symposium on Applied Computing(SAC) 2005*, March 2005.
- [34] EPCGlobal. *Application Level Event (ALE) Specification Version 1.0*, September 2005.
- [35] EPCGlobal. *EPCglobal Architecture Framework Version 1.0*, July 2005.
- [36] EPCGlobal. *Object Naming Service (ONS) Specification Version 1.0*, October 2005.
- [37] 山原裕之, 島川博光. 個人の習慣的行動における半順序関係に注目した振る舞い認識手法. **情報科学技術フォーラム (FIT2005)**, pages 271–272, September 2005.

付 録 A 実験用コンテキスト設定ファイル

動作検証で用いたコンテキスト設定ファイルを以下に載せる.

```
<?xml version="1.0" encoding="shift_jis"?>
<combi>
  <event>
    <epc>800007B80000FA00000000001</epc>
    <name>携帯電話 </name>
  </event>
  <event>
    <epc>800007B80000FA00000000002</epc>
    <name>財布</name>
  </event>
  <event>
    <epc>800007B80000FA00000000003</epc>
    <name>鍵</name>
  </event>
</combi>
<combi>
  <event>
    <epc>800007B80000FA00000000004</epc>
    <name>靴</name>
  </event>
  <combi>
    <event>
      <epc>800007B80000FA00000000007</epc>
      <name>靴</name>
    </event>
    <event>
      <epc>800007B80000FA00000000008</epc>
      <name>Z202 ドア</name>
    </event>
  </combi>
  <combi>
    <event>
      <epc>800007B80000FA00000000009</epc>
      <name>車</name>
    </event>
    <context>context1</context>
  </combi>
  <combi>
    <event>
      <epc>800007B80000FA0000000000A</epc>
      <name>バス</name>
    </event>
    <context>context2</context>
  </combi>
</combi>
</combi>
<combi>
  <event>
    <epc>800007B80000FA00000000005</epc>
    <name>風呂道具</name>
  </event>
  <combi>
    <event>
      <epc>800007B80000FA00000000007</epc>
      <name>靴</name>
    </event>
  </combi>
</combi>
```

```

    </event>
    <event>
      <epc>800007B80000FA00000000008</epc>
      <name>Z202 ドア</name>
    </event>
    <combi>
      <event>
        <epc>800007B80000FA00000000009</epc>
        <name>車</name>
      </event>
      <context>context3</context>
    </combi>
    <combi>
      <event>
        <epc>800007B80000FA0000000000B</epc>
        <name>Δ N ドア</name>
      </event>
      <context>context4</context>
    </combi>
  </combi>
</combi>
<combi>
  <event>
    <epc>800007B80000FA00000000006</epc>
    <name>ノート PC</name>
  </event>
  <combi>
    <event>
      <epc>800007B80000FA00000000007</epc>
      <name>靴</name>
    </event>
    <event>
      <epc>800007B80000FA00000000008</epc>
      <name>Z202 ドア</name>
    </event>
    <combi>
      <event>
        <epc>800007B80000FA0000000000C</epc>
        <name>ι 23 机</name>
      </event>
      <context>context5</context>
    </combi>
  </combi>
</combi>
</combi>

```