

修士論文

2005年度(平成17年度)

コンテキストウェアサービスのための
ルール記述支援機構

慶應義塾大学大学院 政策・メディア研究科

出内 将夫

ide@ht.sfc.keio.ac.jp

コンテキストウェアサービスのための ルール記述支援機構

本論文は、ルールに従って動作するコンテキストウェアサービスに対し、ユーザの要求からルールを提案する機構、QuPen(Query-based rule Processing engine)について述べる。現在、ネットワークに接続された様々なセンサや機器から情報を取得し、ユーザが置かれた状況に合わせて動作するコンテキストウェアサービスが研究されている。このコンテキストウェアサービスを実現する手法として、あらかじめセンサや機器の状態を条件としてサービスを動作させるルールを記述し、サービスを自動化する手法がある。

現在、コンテキストウェアサービスのルールを記述するユーザは、要求するサービスではなく、サービスを動作させる状況について考慮する必要がある。これでは、センサや機器が今後増加した場合、ユーザがルールを記述する際に考慮しなければならないセンサや機器の情報は増加し、ルールの作成が困難になる。QuPenはユーザの要求であるサービスの動作から、サービスを受けていた状況を条件として作成し、ユーザにルールに用いる条件を提案する。

本論文では、まずコンテキストについて定義した後、今後の環境について述べ、QuPenの利用が想定されるサービスについて述べる。次に、ユーザの要求からルールを提案する手法の必要性を述べた後、機能要件を提示する。そして、QuPenの設計および実装について述べた後、本システムを定性・定量の両側面から評価する。最後に、今後の課題について述べ、本論文をまとめる。

キーワード：

- 1 コンテキストウェアサービス 2 ルール自動生成 3 ルールベース
4 ユビキタスコンピューティング 5 決定木

慶應義塾大学 政策・メディア研究科
出内 将夫

Abstract of Master's Thesis

A Rule Description Support Mechanism for Context-aware Services

This thesis explores a rule description support mechanism for context-aware services. Context-aware services currently investigated in the field of ubiquitous computing respond to contextual information gathered through various on-line sensors and devices. A rule-based application is one of the existing approaches to such context-aware applications and operates a variety of services based on conditions of the sensors and devices.

Currently, the user who describes rules for such applications needs to take into consideration not only the type of service they would like to request but also context in which the particular service is run. As such, as the number of sensors and devices multiplies, the user must consider the increasing amount of information while writing the rules. As a solution to this problem, QuPen (Query-based rule Processing engine) is developed. Instead of analyzing context per se, QuPen a) directly examines conditions under which services requested by the user are run, b) lists these conditions in the form of context in which the user received the requested service in the past, and c) presents those conditions to the user.

The first part of this thesis defines "context," describes future ubiquitous environments for context-aware applications, and discusses various services for which support mechanisms such as QuPen will be used. The next part of the thesis argues for the mechanism that suggests rules based on its own analysis of the services requested by the user. It also lists specifications required of such mechanisms. After presenting design and implementation of QuPen, subsequent chapters of the thesis evaluate the system both qualitatively and quantitatively. Finally, the thesis ends by suggesting directions for future research in this line of investigation.

Keywords:

1 context aware services 2 rule auto generation 3 rule-base 4 ubiquitous computing
5 decision tree

Masao Ideuchi

Media and Governance
Keio University

目次

第1章	序論	1
1.1	本研究の背景	2
1.2	問題意識	2
1.3	本研究の目的および意義	3
1.4	本論文の構成	4
第2章	ルールに基づいたコンテキストウェアサービス	5
2.1	用語定義	6
2.2	コンテキストとして用いられる情報	6
2.2.1	人によって記述された情報	7
2.2.2	センサや機器から取得する物理的な情報	8
2.2.3	物理的な情報を加工した情報	9
2.2.4	時間を表わす情報	10
2.3	対象サービス	10
2.3.1	状況との因果関係	11
2.3.2	現状のコンテキストウェアサービス	11
2.3.3	今後のコンテキストウェアサービス	11
2.4	ルールの記述方式	12
2.4.1	現在考えられているルール記述方式	12
2.4.2	現状のルール記述方式の問題点	13
2.5	要求分析	13
2.5.1	ユーザが興味を抱く状況	14
2.5.2	本研究が提案するルール記述方式	14
2.6	既存研究	15
2.6.1	現状行われているコンテキストに関する研究	15
2.6.2	関連研究	17
2.7	本章のまとめ	18
第3章	アプローチ	19
3.1	機能要件	20
3.2	アルゴリズムの比較検討	20
3.2.1	統計的手法	21
3.2.2	ニューラルネットワーク	21
3.2.3	ベイジアンネットワーク	22

3.2.4	決定木学習	23
3.3	決定木による普段の状況モデル	24
3.3.1	木構造	24
3.3.2	決定木における木構造	25
3.3.3	普段の状況を表現する決定木	25
3.4	決定木の構築手法と機能要件に対する妥当性	26
3.4.1	条件として利用する属性の選択基準	27
3.4.2	条件となる属性の値の選択	28
3.4.3	提案する条件の作成	28
3.5	クラス属性の離散化	28
3.5.1	近似性の算出手法	29
3.5.2	適切な閾値の設定	30
3.6	本章のまとめ	30
第4章	QuPen の設計	31
4.1	QuPen の概要	32
4.1.1	ハードウェア構成	32
4.1.2	ソフトウェア構成	33
4.1.3	動作概要	33
4.2	各モジュールの設計	35
4.2.1	情報取得部	35
4.2.2	履歴管理部	36
4.2.3	統計処理部	37
4.2.4	木作成部	37
4.2.5	条件出力部	38
4.3	本章のまとめ	38
第5章	QuPen の実装	39
5.1	実装概要	40
5.2	各モジュールの実装	40
5.2.1	情報取得部	40
5.2.2	履歴管理部	42
5.2.3	統計処理部	43
5.2.4	木作成部	44
5.2.5	条件出力部	45
5.3	本章のまとめ	45
第6章	評価	46
6.1	実験	47
6.1.1	実験概要	47

6.1.2	実験環境	47
6.1.3	実験内容	48
6.1.4	実験結果	49
6.2	定性評価	49
6.2.1	ルールの記述	49
6.2.2	関連研究との比較	50
6.3	定量評価	51
6.3.1	離散値を取るクラス属性	51
6.3.2	連続値を取るクラス属性	52
6.4	本章のまとめ	52
第7章	結論	54
7.1	今後の課題	55
7.1.1	多様なセンサや機器を用いた評価	55
7.1.2	ルールに基づくコンテキストウェアサービス実現フレームワークとの連携	55
7.1.3	離散化の計算量改善	55
7.1.4	ルールの因果関係記述	55
7.2	まとめ	56

目 次

1.1	ルールに基づくコンテキストウェアサービスの概念図	3
2.1	コンテキストウェアサービスの定義	7
2.2	現在のセンサ利用形態	8
2.3	今後のセンサ利用形態	9
2.4	ユーザがコンテキストを記述する際に考慮すべき情報	13
2.5	本研究が提案するルール記述方式	15
3.1	ニューラルネットワークの例	22
3.2	ベイジアンネットワークの例	23
3.3	木構造の例	24
3.4	決定木の例	25
3.5	扇風機の電源状態を興味ある属性に指定して作成した決定木	26
3.6	デンドログラムの例	29
4.1	ハードウェア構成図	33
4.2	ソフトウェア構成図	34
4.3	情報蓄積作業シーケンス図	35
4.4	条件作成作業シーケンス図	36
5.1	情報取得部クラス図	41
5.2	履歴管理部の出力データサンプル	42
5.3	属性ファイル例	43
5.4	決定木を表現する XML の出力結果例	44
6.1	実験環境の見取り図	48
6.2	実験結果として得られた決定木	49
6.3	実験結果として得られたルール	50
6.4	クラス属性が離散値を取る場合に条件出力にかかる時間	52
6.5	クラス属性が連続値を取る場合に条件出力にかかる時間	53

表 目 次

3.1	条件自動算出手法とその比較	20
3.2	木構造，決定木，定常状態を記述した木の対応表	27

第1章 序論

本章では，本研究の背景である，ユビキタスコンピューティング環境について述べる．その後，本研究の目的および意義を述べ，最後に本論文の構成について述べる．

1.1 本研究の背景

近年、情報技術の進歩により、計算能力やネットワーク接続性を持った様々なデバイスの小型化や低価格化が進んでいる。これらのデバイスが遍在し、ネットワークを通して互いに協調動作することで、人々の日常生活を支援する環境をユビキタスコンピューティング環境 [43] と呼ぶ。現在様々な施設がユビキタスコンピューティング環境の実験施設として構築され [26][30][42]、様々な実証実験 [23][31] も行われている。今後、このような情報環境が、オフィスやキャンパス、家庭や公共空間へと浸透していくと考えられる。

上述の環境において、ユーザの個人情報を用いたり、センサや機器の利用状況を元に、人々の行動に応じて動作を変化させる、コンテキストウェアサービスの研究が行われている [12][36][25]。例として、会議中は携帯電話を自動保留にするサービス [39] や、公共端末においてユーザの行き先に応じてナビゲーションを行うサービス [35] が挙げられる。このようなコンテキストウェアサービスを実現するための手法として、サービスをどのような状況で提供してほしいかをルールとして記述し、そのルールに従ってアプリケーションを動作させる方法がある [46][52]。例えば、ユーザが部屋に入ると自動的に部屋の電灯をつけ、エアコンの電源を入れ、テレビをつけるなどのサービスが考えられている。図 1.1 にルールに基づくコンテキストウェアサービスの概念図を示す。

環境の差異により、同じルールから違う結果が得られることもあるため、サービスを受けるユーザ自身がサービス利用環境に合わせてルールを記述し、上記のようなコンテキストウェアサービスを実現する方法が考えられている。また、そのためにユーザが理解しやすい記述方法の研究がされている [6]。ユーザがルールを作成するためには、目的に合わせてアプリケーションを動作させる条件となるコンテキストを指定する必要がある。前に述べた例を用いた場合、会議中と判断するためにユーザが持つ持たせた超音波タグから部屋にいるメンバーやメンバー数を条件にしたり、ユーザが部屋に入ったことを検知するために赤外センサを用いる方法が挙げられる。これらの条件には、例に挙げた以外のセンサを用いた実現方法が考えられ、ユーザは適したセンサを選別して利用する必要がある。

1.2 問題意識

ユーザがコンテキストウェアサービスに用いるルールを記述するためには、サービスに必要なアプリケーションを動作させたい状況について考え、ルールとして記述するコンテキストとして適切な情報を選別し、閾値や条件値を設定する必要がある。ルールとして記述する情報が少なく、条件も単純なルールを作成することは比較的容易であると考えられるが、ルール通りにアプリケーションが動作してもユーザの意図が十分に反映できない状況が存在する。この原因としては、環境による限界と、ユーザによる限界が考えられる。前者は、コンテキストとして利用できる情報が絶対的に

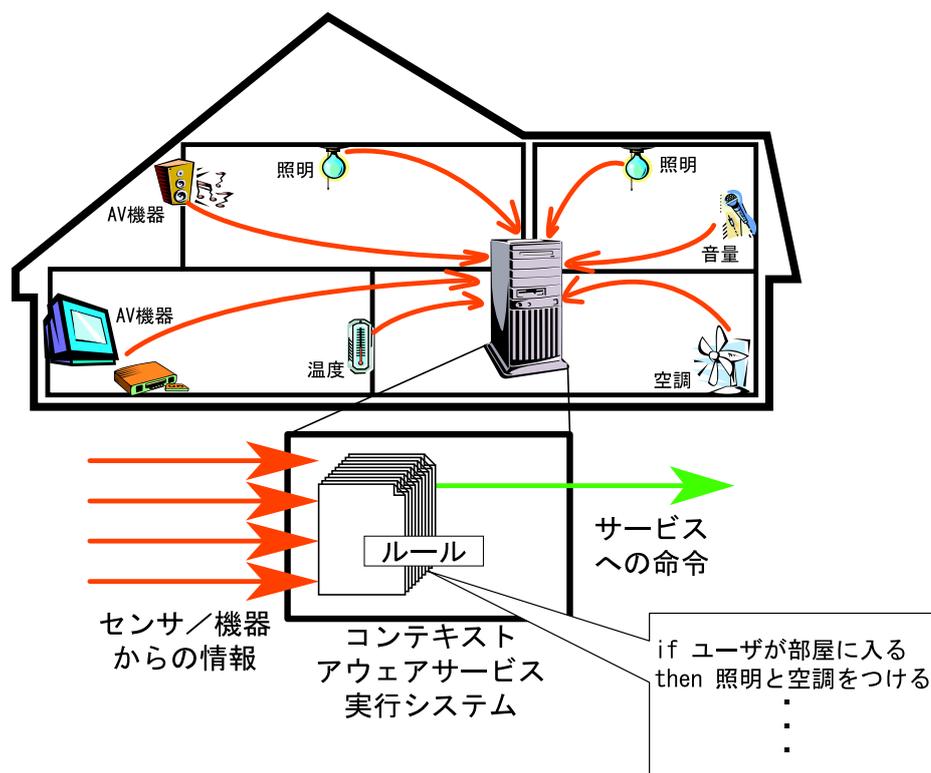


図 1.1: ルールに基づくコンテキストウェアサービスの概念図

不足していて十分に記述できない場合であり，後者はユーザがルールに記述した情報や条件が不適切な場合である．

ユビキタスコンピューティング環境では，環境に存在するセンサの数も多くなり，かつネットワーク上から情報を参照可能になると考えられる．このため，コンテキストを記述するための情報が十分に得られる環境が整い，環境による限界は解消していくと考えられる．しかし，情報の数が多くなればなるほど，ルールとして記述できる情報，作成できる条件の組み合わせも膨大となり，適切なルールをユーザが選別し，条件を作成する作業は困難となるため，ユーザによる限界がルール記述の問題点となる．

1.3 本研究の目的および意義

本研究では，ユーザがアプリケーションを動作させたい状況を指定し，その状況を判別する決定木を作成する機構，QuPen を構築する．これにより，アプリケーションが動作すべき状況の記述に必要な情報を抽出し，ユーザが行うコンテキストウェアサービスのルール記述作業を支援する．QuPen はセンサからの情報や機器利用状態の履歴を用いて，ルールを記述する上で適切なセンサや機器の選出，判定基準の設定を行い，結果をユーザに提示する．ユーザはその情報を元に，意図に即したルールを選

別し，コンテキストウェアサービスのルールとして利用できる．これによりユーザがルールを記述する際の適切な情報の選出作業，適切な条件の記述作業の手間が軽減できる．

1.4 本論文の構成

本論文は，本章を含め全 7 章から構成される．次の第 2 章で本研究が対象とするアプリケーションと，アプリケーションが必要とする情報についてまとめ，第 3 章で本研究がアプリケーションに提供する情報を抽出する手法について述べる．そして第 4 章では，本研究で作成する QuPen の設計について，第 5 章で実装について述べる．さらに第 6 章で QuPen の動作実験と評価について述べ，第 7 章で本論文をまとめる．

第2章 ルールに基づいたコンテキスト アウェアサービス

本章では，ユビキタスコンピューティング環境で用いられるルールを用いたコンテキストアウェアサービスについて述べ，本研究の対象領域を明確にする．まず，本研究で用いる用語について整理し，それぞれの関係について述べる．次に，条件として記述されるコンテキスト，利用されるサービス，ルールの記述方法の3つについて，現状と今後について考察する．そして，本研究が提案するルール記述方式について述べる．最後に関連研究を挙げ，本研究との位置づけを明確にし，本章をまとめる．

2.1 用語定義

本節では，本論文が対象とするルールに基づいたコンテキストウェアサービスを明確にする．そのために，まずコンテキスト，ルールについて定義し，次にそれらを用いてルールに基づいたコンテキストウェアサービスを定義する．

コンテキスト

本論文で用いるコンテキストとは，情報の種類とその情報がある値の一つ以上の集合を指す．情報の種類の例として，温度や部屋にいるユーザ，テレビの電源状態などが挙げられる．これに対応する情報の値の例としては，それぞれ 24 度や 30 度などの数値，部屋にいるユーザ名のリスト，ON や OFF となる．本研究が扱う情報の種類は 2.2 で詳しく述べる．情報の種類を Att (Attribute)，情報の値を Val (Value) として表した場合， Ctx (Context) は， Att と Val の組み合わせが 1 つ以上集まった集合として表現し，以下の式で定義される．

$$Ctx = \{ \langle Att_i, Val_i \rangle \mid (i \geq 1) \} \quad (2.1)$$

ルール

本研究で用いるルールとは，条件となるコンテキストの集合と，結果として起動されるアプリケーションの 2 つの要素を持ち，コンテキストが満たされた場合にアプリケーションを実行するという命令の記述である．条件にはコンテキストが持つ情報の属性それぞれに対して，情報の値を比較する式が存在する．例として，現在の値を Val_{now} ，ルールに記述された条件に記述された値を $Val_{condition}$ とすると，情報の値が数値の場合は $Val_{now} > Val_{condition}$ (条件値より大きい) や $Val_{now} < Val_{condition}$ (条件値より小さい) などの比較，離散値あるいは離散値のリストの場合は $Val_{now} == Val_{condition}$ (条件値と一致) や $Val_{now} \subset Val_{condition}$ (条件値を含む) などの比較が考えられる．

ルールに基づいたコンテキストウェアサービスとは，上述したルールによって動作するアプリケーションが 1 つ以上集まって提供されるサービスを指す．例えば，ユーザが椅子に座ったら，ディスプレイの電源がつき，音楽が流れ始めるなどのスマートオフィスサービスが考えられる．以上で述べたコンテキスト，ルール，アプリケーション，そしてコンテキストウェアサービスの関係を図 2.1 に示す．

2.2 コンテキストとして用いられる情報

本節では，ルールを用いたコンテキストウェアサービスに利用されるコンテキストについてまとめる．まず，ユーザが入力した情報を用いるコンテキスト，センサから取得した情報を用いたコンテキストについて述べ，本研究との位置づけを述べる．

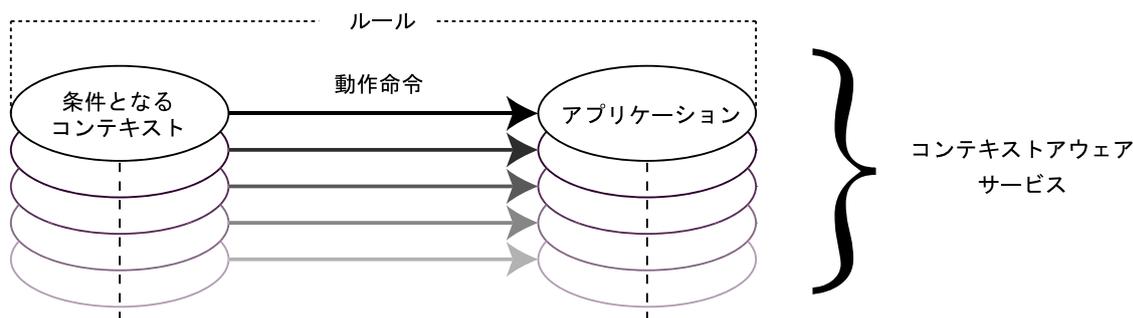


図 2.1: コンテキストアウェアサービスの定義

2.2.1 人によって記述された情報

この情報は、主に公共空間を中心とした不特定多数のユーザが利用するサービスにおいて、コンテキストアウェアサービスを構築する際に条件として用いられている。この情報の例として、ユーザの個人情報やユーザの意図などが挙げられ、性別や年齢に応じた広告を提供するサービスや、ユーザが行きたい場所や到着したい時間に応じた案内を提示するサービスに利用される。これらの情報を、長期に渡って変化しない静的な情報と短期的期間のみ有効な動的な情報に分類し、以下に述べる。

静的な情報

先に述べた、年齢や性別などの個人情報が静的な情報に該当する。この情報は位置情報センサや画像センサなどで ID を認識し、同じユーザであることが分かれば、一度入力された情報を再利用できるため、次節で述べるセンサの情報から認識可能となる。しかし、過度に個人情報を要求することはプライバシーが問題となるため、サービスの信頼度に応じたプライバシーの提供手法が研究されている [41][28]。

動的な情報

先の例で、行き先や到着時刻などユーザの意図を表す情報は動的な情報といえる。これらの情報は予定表やメールとして、予めユーザが記述しておいた情報から利用する方法も考えられており、ユーザの ID を元に情報を取得することも今後可能になっていくと考えられる。

この情報は、人が記述する際の表記のゆれや、記述された情報の意味などを考慮した処理を行わないと、同じコンテキストが多種多様な形で表現されてしまうため、扱いが難しい。また、本研究で対象とするのは家庭やオフィスなどの屋内の環境であるため、今回は対象としない。今後、人によって記述された情報が意味解析などを用いて統一的に表現される技術が整えば、本研究でもコンテキストとして利用可能である。

現在、ユーザの意図を読み取れる情報として予定表の項目が標準化されている [45]。今後、こういった記述方式の標準化や表記のゆれを補正する技術 [51] や、単語の意味

を解釈する技術 [18] が発展していけば，これらの情報を基にルールを記述するサービスが実現できると考えられる．

2.2.2 センサや機器から取得する物理的な情報

ユーザが存在する場所や，周囲の環境の状態を元にサービスを変化させたい場合に，環境に設置あるいはユーザが持つセンサや機器から取得した情報がコンテキストとして用いられる．例えば，ユーザの入室を画像センサや人感センサ，あるいはタグ認識技術などによって取得し，部屋の機器の電源が OFF であれば ON にすることが考えられる．これらの情報を用いることで，ユーザが明示的に指示を出さずとも，ユーザは適切なタイミングでサービスを受けられる．

本研究では，主にこの情報を扱う．次節でサービスが今後どのように発展するかを考察するため，その基盤となるセンサの利用形態について以下にまとめる．

現在のセンサ利用形態

家庭やオフィスなどを中心に，人々の身の回りには既に様々な形でセンサが設置されている．しかし，現状で設置されているセンサは，後述するサービスと密に繋がった形で目的に応じて設置され，他のサービスから利用できない．例えば，防犯用の侵入検知サービス [37] では，ドアや窓の開閉状態や人感検知センサを設置し，サービス独自のネットワークを構築しているが，その情報が住人の帰宅を検知して他のサービスを提供することには用いられていない．現在のセンサ利用形態を図 2.2 に示す．



図 2.2: 現在のセンサ利用形態

今後のセンサ利用形態

今後は、前述した目的別に設置されたセンサがネットワークからアクセス可能になり、他のサービスからも利用可能となると考えられる [47]。現在、国内外で Mote[21] や Smart-Its[9]、U-cube[48] など、多種のセンサを搭載したチップの小型化、低価格化が進んでいる。これらのハードウェア技術の進歩や最初に述べたインフラが整うことで、ユーザが必要に応じてセンサを設置し、ネットワークに組み込める環境も構築されていくと考えられる。今後のセンサ利用形態として予想される環境を図 2.3 に示す。

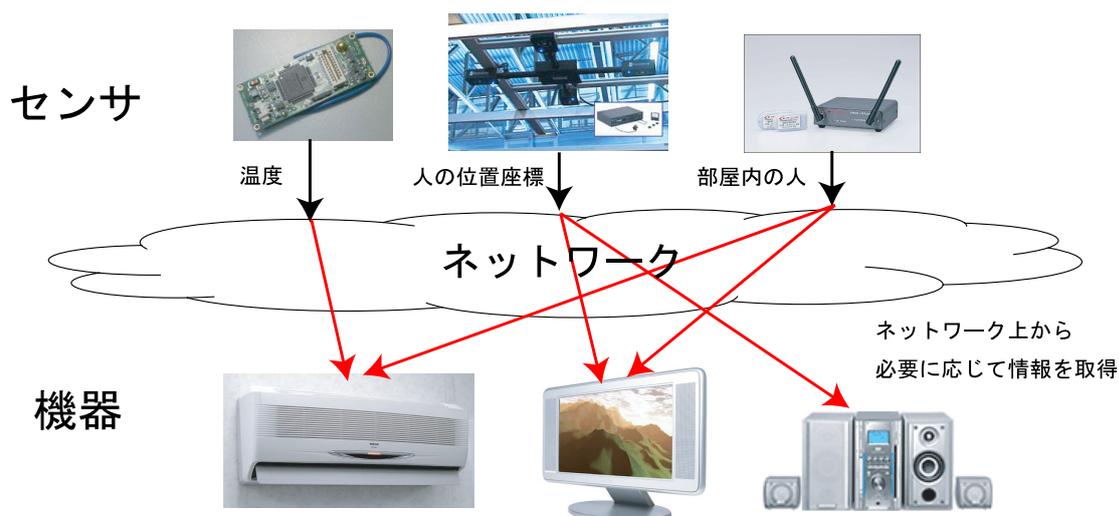


図 2.3: 今後のセンサ利用形態

今後センサを管理する技術として、センサやセンサデータを記述する方式の標準化や、ネットワーク上から必要なセンサを探し出す技術が発展すると考えられる。これらの技術は現在、屋外に散布し、自然観測や生態観測に用いるセンサに対して整備され始めている [27][3]。

2.2.3 物理的な情報を加工した情報

前項で述べたセンサから取得した情報に対して、一つ以上の情報を用いて加工して得られる情報を指す。時間的連続性や空間的連続性を考慮して加工した情報や、既存のコンテキスト抽出機構を用いて環境やユーザの状態を分類した情報などがこれに当たる。それぞれの例について以下にまとめる。

時間的連続性を考慮して加工した情報

ある部屋でのユーザ A の滞在時間、サービスが継続して利用された時間などを指す。ルール条件として用いるコンテキストの例としては、ユーザが部屋に滞

在する時間がコンテキストによって分類できれば，ユーザが部屋にいる間にコーヒーを淹れるサービスや，部屋を出る時間までに風呂の準備を整えておくサービスなどに応用できる．

空間的連続性を考慮して加工した情報

単一の位置情報センサで部屋全体をカバーできない場合，複数の位置情報センサ A, B, C を組み合わせることで，部屋全体をカバーでき，部屋という意味のある単位でユーザの滞在を取得できる．他にも部屋の温度や照度といった，同じ空間内にあるセンサを連携させて，ある空間としての状態を取得し，コンテキストとして利用できる．

既存のコンテキスト抽出機構を用いて加工した情報

現在，ユーザの位置情報や，身の回りのセンサの情報から，ユーザが食事中，休憩中，読書中などといった行動を分類する研究 [7] が行われている．そのようなシステムからの出力結果を，一つの情報として扱い条件作成に用いるコンテキストとして利用できる．

本研究では，前項で述べた情報に加えて，この加工した情報も扱う．主に時間的連続性，空間的連続性を考慮した加工や，位置情報センサから取得した ID をユーザ名に置き換えるなどの加工したデータを扱う．

2.2.4 時間を表わす情報

以上で述べた情報に対して，いつ取得された情報かを示す指標となる情報である．例として，午前中，午後といった時間帯や曜日，季節など，現在の時刻から得られる情報が挙げられる．この情報を活用することで，時間帯や日時に影響を受けて利用形態が変化するサービスをルールとして記述できる．

本研究では，センサから得られる情報や機器の利用状況を中心として条件に記述するコンテキストの抽出を目的とするが，必要に応じてこの情報も考慮に入れる．

2.3 対象サービス

本節ではまず，ルールを基にしたコンテキストウェアサービスが対象とする，状況と因果関係が強いサービスについて述べ，どのような種類のサービスが対象となるかについて述べる．次に，現在利用されているコンテキストウェアサービスと今後利用されると予想されるコンテキストウェアサービスについて述べる．

2.3.1 状況との因果関係

ルールを基にしたコンテキストウェアサービスが対象とするのは、「ある状況になった場合にこのサービスを利用したい」といった、状況との因果関係が非常に強いサービスである。そのようなサービスとして、以下のような種類のサービスが挙げられる。

日々利用しているサービス

起床後に部屋のライトを点灯し、エアコンのスイッチを入れる、帰宅したら風呂を沸かす、などのユーザが日々繰り返し利用しているサービスや、夕食後にソファに座ってテレビをつける、デスクに座ったら音楽を流し始めるなどといったユーザが習慣づけて利用しているサービスを指す。

監視サービス

ユーザが外出する時に忘れ物チェックをするサービスや、ユーザが家を出ている間に、家庭内のペットや老人の状態をセンサから得られる情報や機器の利用状態を監視して、異常があると認められる場合、メールで通知するといったサービスが該当する。

2.3.2 現状のコンテキストウェアサービス

現在、ルールに基づいたコンテキストウェアアプリケーションとして、以下に述べる様々なアプリケーションが実用化されている。単体のセンサを利用した例として、ドアの開閉や照明の自動点灯サービスが、複数のセンサを利用した例としては、盗難検知や侵入検知、老人の見守りなどの危機的状況を監視するサービスなどが挙げられる。これらは 2.2.2 で述べたように、サービスごとの要求に応じてセンサが設置されている。このようなサービスで、センサ単体を利用する場合は導入に必要なコストは少なく済むが、センサを様々な場所に設置して複数連携させて利用する場合、サービス提供者側が利用者の環境に合わせてセンサの設置や設定、ネットワークの構築を個別に行う必要があるため、導入に必要なコストは高価となる。そのため、そのコストに見合う、盗難検知や侵入検知、老人の見守りなどの危機的な要求があるサービスに用いられている。

2.3.3 今後のコンテキストウェアサービス

今後 2.2.2 で述べたように、機器に埋め込まれたセンサがネットワーク化されることで、既存のセンサが利用可能になり、サービスが独自にネットワークを構築する必要はなくなる。また、ユーザが手軽に設置しネットワークに組み込めるセンサが登場することで、様々な場所に設置されたセンサを複数連携させて、より詳細なコンテキストの識別を必要とするサービスを導入するコストも安価になると考えられる。今後このようなセンサを利用して、鍵の閉め忘れや忘れ物の検知といった備忘サービスや機器制御の自動化など日用的なサービスが発展していくと考えられる [8]。

2.4 ルールの記述方式

ルールとは，2.1 で述べたように，あるコンテキストになった場合に，あるアプリケーションに動作命令を出す記述である．このルールに記述されるコンテキストには2.3.1 で述べた，同じ作業を行う状況のコンテキストが条件として記述され，その作業を補助するサービスが構成されるアプリケーションと結び付けられる．

本節では，まず現状でルールを記述する作業が現在どのように行われているかについて述べ，その問題点について触れる．その後，本研究が対象とする2.2.2 や2.3.3 で述べた環境下で必要と予想される記述方式について述べる．

2.4.1 現在考えられているルール記述方式

現在2.2.2 と2.3 で述べた環境下で考えられているルールの記述方式として，埋め込み型とユーザ記述型の2種類が存在する．以下にそれぞれについて説明する．

埋め込み型

現在，2.3.2 で述べたサービスに利用するルールは，既にサービスに埋め込まれていてユーザが操作できない形で存在している．これは，2.2.2 で述べたように，センサが一つのサービスのみを目的として設置されている環境の特徴である．

ユーザ記述型

また現在，ユーザがコンテキストをルールとして記述し，機器の操作を自動化するフレームワークが研究段階として考えられている [46][52]．その際，ユーザは以下の手順でルールを記述する．

1. サービスを自動化したい状況について想像する

2.3.1 で述べた日々利用しているサービスを対象とする場合は，普段サービスを利用している状況が該当する．その状況が複数存在する場合は，それを全て洗い出す．監視サービスに対しては，監視して欲しい対象がどのような状況であるかを想像する．

2. コンテキストとして記述する

サービスを自動的に動作させる状況がどのようなセンサや機器を基に記述できるかを考え，センサの値や機器の動作状況を2.1のルールで述べた条件値として設定し，コンテキストとして記述する．

3. サービスと結びつける

サービスを構成するアプリケーションに対して、動作命令を記述する。複数のアプリケーションを連携させる必要がある場合は、連携を考慮して記述する必要がある。

2.4.2 現状のルール記述方式の問題点

現在は、ユーザがルールを記述する際、センサや機器は全てユーザが把握しており、センサの値に対する閾値や機器の動作状況などの条件設定も手動で行えるという想定でコンテキストウェアサービスが考えられている。しかし今後は、2.3.3で述べた環境が整うにつれ、センサの種類や数が増えていくと予想される。そのため、上述した作業のうち、2番目に挙げたセンサや機器の選別、適切な条件値の設定を全て行うことが難しくなる。この記述方式の問題点を概念図として図2.4に示す。

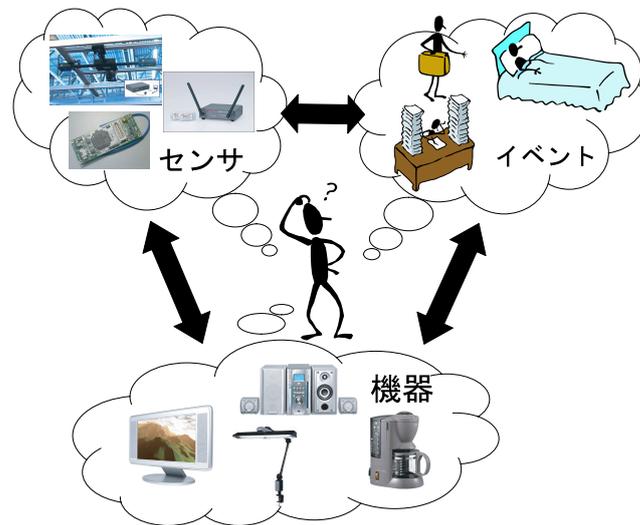


図 2.4: ユーザがコンテキストを記述する際に考慮すべき情報

2.5 要求分析

本節では、条件記述の際にユーザが興味を抱く状況について分析する。そして、2.4.2で述べたルール作成時の問題点を、本研究が提供するルール提案手法でどのように解決するかについて述べる。

2.5.1 ユーザが興味を抱く状況

まず，ユーザが興味を抱く状況とは何かについて明確にする．コンテキストウェアサービスのルールを作成するためには，サービスを利用したい状況のコンテキストをルールの条件として記述する必要がある．このサービスを利用したい状況がユーザの興味を抱く状況に該当する．2.3.1 で述べたサービスに対して，サービスを利用したい状況について分析する．

日々利用しているサービスの自動化

日々利用しているサービスをコンテキストウェアサービスとして自動化するためには，普段サービスを利用している状況をコンテキストとして記述し，ルールの条件として設定する必要がある．この場合，普段利用するサービスで動作させている機器の利用状態が情報として取得できれば，普段サービスを利用している状況のコンテキストは「機器が動作しているとき」として指定可能である．しかし，このままルールに記述してしまえば「機器が動作している状態であれば，機器を動作させる」となり，ルールとしての意味を持たない「機器が動作している状態であったとき，他のセンサや機器は A, B, C, \dots という状態であった」というコンテキストが普段の状況から逆算できれば，ルールとして記述できる．

監視サービスの自動化

監視サービスは監視対象が普段どんなコンテキストであるか判明すれば，そのコンテキストから外れた場合通知する，といった形でルール作成の際に参考できる．具体的には，老人の転倒を検知したい場合，ユーザは転倒がどの情報を用いれば識別が可能かを推測できる．老人に心拍や血圧などの生体センサがつけられている場合，その情報が参考になるであろう，生体センサがない場合は同じ部屋にどのくらいの時間滞在したか，あるいは老人がいる部屋の機器が最後に操作されてからどのくらいの時間が経過したか，という情報が参考になるであろう，といった推測である．それらの情報が普段どのような値を示しているか判明すれば，その値を基準として通知する値を設定できる．

2.5.2 本研究が提案するルール記述方式

前項で述べた内容から，ルール作成の際に普段の状況がコンテキストとして記述できれば，参考として利用できるといえる．本研究では，ユーザが興味を抱いた状況を指定するだけで，その状況を識別するために必要なセンサや機器の選別，適切な条件値の設定を提案してくれるシステムを構築し，以上の問題点を解決する．これにより，ユーザが考慮すべき状況は2.4.1のユーザ記述型で述べた1番目の状況だけとなり，多くのセンサや機器，またそこから得られる情報を想定する必要がなくなり，条件を記述する手間が省ける．図2.5に，本研究が提案するルール記述形式の概念図を示す．

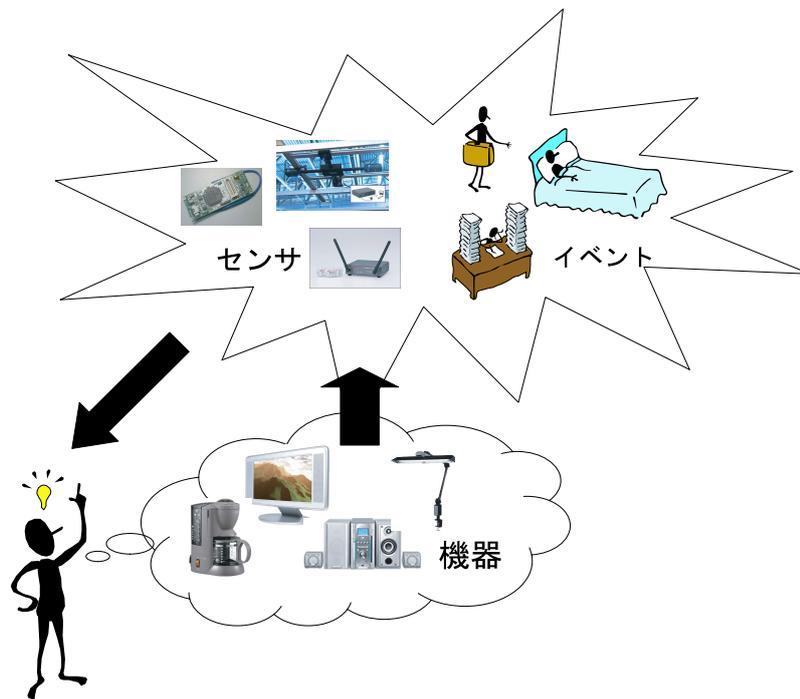


図 2.5: 本研究が提案するルール記述方式

普段の状況は，ユーザや環境ごとに差異があると予想されるため，コンテキストの履歴を用いたモデル化が必要となる．モデル化の手法として，統計や学習アルゴリズムが利用できる．モデル化手法については，次節で関連研究で利用されている手法について挙げた後，次章で検討結果について述べる．

2.6 既存研究

本節では，コンテキストウェアサービスを対象としたシステムについて述べる．まず，本研究が対象とするコンテキストウェアサービス提供フレームワークについて述べ，現行の研究でセンサや機器からどの程度のコンテキストが取得できているかについて述べる．次に，ルール記述の支援を目的とした関連研究について述べる．

2.6.1 現状行われているコンテキストに関する研究

本項ではまず，本研究が前提とするコンテキストウェアサービスを実現するフレームワークについて述べる．次に，本研究が想定するセンサや機器がネットワーク化された環境において，どの程度のコンテキストが取得できると考えているかについて，現状行われているコンテキスト解析の研究を例示する．

iCAP

Intel Research[1] による iCAP[15] は、ルールに基づいたコンテキストウェアサービスを構築する手法として、ユーザが簡単にルールを生成できるビジュアルプログラミング環境を提供するシステムである。ユーザはルールの条件となるデバイスを input として、結果として動作させるアプリケーションを output として登録する。その際、ユーザ自身が簡単な絵を描いてアイコンとして登録することで、ユーザが理解しやすいビジュアルプログラミング環境を実現している。

コンテキストウェアサービスに対して、ユーザが簡単に記述できる環境を提供する、という部分は本研究と目的が同じである。また、センサや機器が接続可能なネットワークが環境に構築され、コンテキストウェアサービスが実用段階に入った直後は、ユーザがセンサや機器を把握し、システムにセンサや機器を登録する作業も行うと想定されるため、このような機構は必要であると考えられる。しかし、本研究が想定するように、センサや機器の数が増えて行くと、いずれ把握できる数に限界が来ると考えられるため、ビジュアルプログラミングのみで全てのコンテキストウェアサービスが実現できるとは考えにくい。本研究とここで述べた研究の関係としては、提案するルールを表現する際やユーザによる編集を行う際に、このようなシステムを用いることで、ユーザがルールを理解、編集することが容易になると考えられる。

乳幼児の事故予防

産総研 [5] による乳幼児の事故予防のための知識データベースと事故予測モデル [53] は、子供に 3 次元の位置を取れるセンサを装着し、子供の行動からベイジアンネットワークを構築してモデル化することで、高熱を発する機器に触れてやけどをしたり、高いところの物を自分の頭上に引きずり落として怪我をしたりする、子供に及ぶ危険を回避するシステム構築している。学習対象のデータとして、センサや機器から得られるデータのみでなく、子供の年齢情報や立てるか立てないかなどの 2.2.1 に該当する情報も扱い、情報が欠けていても動作する推論アルゴリズムを利用している。この研究もサービスの部分として、子供が危険と考えられる状況になった場合に親に連絡する、というルールで記述しているサービスを対象としている。

本研究では、ルールに基づくコンテキストウェアサービスを対象としているため、この研究と目的が異なるが、子供の危険度という抽象的なコンテキストを、センサや機器の状態を組み合わせで扱っている部分は興味深い。今後、センサや機器の数が増え、人が入力したデータを用いれば、ルールを用いてこの研究のようなメタなコンテキストも定義可能になると考えられる。この研究の位置づけとしては、ルールの条件としてこの研究の結果として得られたコンテキストを利用できる、といった補助的な位置づけと、ネットワークに接続して複数のデバイスを用いれば、この研究のような抽象度が高いコンテキストが得られるというコンテキスト記述力が増加する、といった本研究の背景としての意味合いがある。

House_n Activity Recognition

MIT(Massachusetts Institute of Technology)[2] による Activity Recognition from User-Annotated Acceleration Data[7] は、ユーザが体に装着した 5 つの加速度センサを用いて、ユーザの行動を分類するシステムを構築している。まず、サンプリングした 5 つの加速度センサからのデータにユーザ自身が 20 種類のラベル付けを行い、次に加速度センサのゆれを波形処理や統計処理にかけ、決定木学習やベイジアンネットワークを用いて各ラベルの特徴を学習させる。そして、新たに取得したデータを学習済みの決定木やベイジアンネットワークにより、自動的にそれぞれのラベルに分類する。結果として、行動 20 種類中 16 種類の分類が 80% 以上の精度で、さらにそのうちの 8 種類は 90% を超える精度で行える。

本研究では、ネットワーク接続性を持った複数種類のセンサや機器を対象として、モデル化を行うため、この研究とはモデル化の対象が異なる。しかし、加速度センサ 5 つのみという限られた状況の中から、高精度でコンテキストを抽出している技術は、次章で述べる本研究で用いる手法を選別する際に参考となった。加速度に特化した解析も行っているため、本研究で構築するシステムを考える際に、センサ独自の解析手法を組み込めるよう考慮した設計を心がけた。今後、この研究のような形で、センサそれぞれに特化した解析手法も考案され、高精度かつ抽象度の高いコンテキストが検知できるようになると考えられる。

2.6.2 関連研究

コンテキストウェアサービスを対象としてコンテキストの履歴を用いてユーザに自動化を提案している既存研究を挙げる。それぞれに対して概要を述べ、本研究との差異を明らかにする。

SmartMacro

名古屋大学の SmartMacro システム [32] は、ユーザが情報機器を操作した履歴から、次に利用するサービスを予測し提案するシステムである。提案結果として選ばれる履歴の判断は、直前の機器操作と現在のコンテキストを元に行われ、過去同じような状況があれば、その際に行った次の機器操作が提案される。また、ユーザは提案された結果を選択して実行するという 2 アクションで機器操作を実行できるインターフェースも備えている。

SmartMacro は、コンテキストの履歴を用いてサービスを監視し、ユーザの要求に応じてサービスの自動化を提案するという点で本研究と似ている。本研究との相違点は、サービスが連続して利用される状況に特化している点である。

Synapse

東京大学の Synapse システム [49] は、サービスを利用する前に発生するイベントを監視し、履歴からイベント系列とサービスの関連性を見出し、サービスの自動化をユーザに提案するシステムである。イベントはコンテキストの変化によって定義されている。

Synapse はコンテキストの履歴を用いてサービスを自動化するという点において、本研究と似ている。本研究との相違点は、コンテキストの変化を表すイベントに着目している点と、ユーザの要求によらず、ある瞬間のコンテキストに対して履歴から確率的に見て実行されるべきと判断されたサービスがユーザに提案される点である。

2.7 本章のまとめ

本章では、本研究の対象となるルールに基づいたコンテキストアウェアアプリケーションとその構成要素について述べ、今後のコンピューティング環境がどのように変化することを想定し、本研究の対象領域について述べ、既存研究との位置づけを明確にした。次章では、本研究が用いる手法について述べる。

第3章 アプローチ

本章では，本研究で用いるユーザが興味を抱いた状況をモデル化する手法について述べる．まず，前章までに述べた目的や要求を満たすための機能要件についてまとめる．そして，利用するアルゴリズムの検討を行い，本研究で利用する決定木の利用方法について述べる．

表 3.1: 条件自動算出手法とその比較

手法	情報依存関係抽出能力	条件設定能力	ルールとの親和性
統計的手法		×	
ニューラルネットワーク		×	×
ベイジアンネットワーク			
決定木			

3.1 機能要件

本節では、次節で行う履歴として与えられる情報をモデル化する手法についての比較を行う際に、評価基準となる機能要件について述べる。統計手法、学習アルゴリズムに求められる必要な機能として、以下の3つが挙げられる。

情報の依存関係抽出能力

履歴として与えられた多種の情報から、どの情報が条件として利用できるかを選別する必要がある。

条件設定能力

上記で得られた条件に利用する情報が、どのような値を取ればユーザが興味を抱いた状況なのかを指定する必要がある。情報の値が数値で表現される場合、適切な閾値を設定する必要がある。

ルールとの親和性

興味を抱いた状況がどの情報を用いてどのような条件で識別できるかを履歴から算出した際に、それをルールとして記述するのが容易かどうかを考慮する必要がある。

3.2 アルゴリズムの比較検討

前節で挙げた要件を考慮し、ユーザの行動をモデル化するために適したアルゴリズムについて検討する。検討したアルゴリズムは、数量化法アプローチ [20]、ニューラルネットワーク [22]、ベイジアンネットワーク [14]、決定木学習 [50] の4つである。検討した結果、決定木学習を用いた。検討結果のまとめを表 3.2 に示す。

以降、情報の種類と情報を属性と値と呼び、値の種類を離散値と連続値に区別して扱う。連続値とは、連続的に値が変化する数値で表現でき、値が数値的に近ければ近い意味を持つ値である。離散値とは、非連続的に値が変化し、値が少しでも異なると意味が全く異なる値である。

3.2.1 統計的手法

ある 1 つの属性が他の属性とどの程度相関があるかを，数式化することで表現する．前者の属性は応答変数と呼ばれ，後者の属性は説明変数と呼ばれる．結果は，応答変数をそれぞれの説明変数に係数を持つ項の総和として表現する数式として得られ，各項の係数が相関の正負を表す．統計的手法を用いる場合は，応答変数としてユーザが興味を持つ状況を表す属性を指定し，説明変数にルールの子の候補とする他の属性を指定する．この数式を 3.1 に示す．

$$y = a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n = \sum_{i=0}^n a_ix_i \quad (3.1)$$

上記の数式を履歴に対して適用するとデータ全体から評価した全体的な相関を表すため，応答変数と説明変数間に限っては情報の依存関係を記述できている．しかし，複数の条件を記述するためには，説明変数間の依存関係も考慮する必要があり，再度数式化が必要となるため，単一のモデルで複数の条件を記述することは困難である．例えば，ある場所に滞在した時間から，老人が倒れたことを判断したい場合，場所によって異常と判断される滞在時間が影響を受けたり，他の属性を用いないと判定が困難な場所も存在すると考えられる．このような場合，ある説明変数で分類した後の履歴に対して，もう一度相関を求める計算を行う必要がある．

また，全ての属性を連続値として扱うため，属性が離散値を取る場合，属性が取る値と同じ数のダミー変数として数式を表現するため [54]，変数の数は膨大になる．さらにその結果は，それぞれの属性を現す変数に係数を持つ項の総和を表す数式で表現され，ある属性がどの値を取った場合にユーザが興味を持つ状況といえるかを判別できないため，条件の設定が難しい．ルールとして記述できるかに関しては，離散値を変換したダミー変数のみであった場合は，相関係数の値を基に条件として解釈することはできるが，連続値を含む場合困難である．

3.2.2 ニューラルネットワーク

入力と出力の属性間の写像関係を表現するグラフ構造を構築する．ニューラルネットワークの例を図 3.1 に示す．ネットワークのノードとして表現するために，全ての属性を一度数値化する必要がある．また，入力，出力として設定する実際の属性以外にも，中間ノードを必要に応じて設定する必要がある．優れたネットワークを構築するためには，経験的に中間ノードを配置することが重要となる．ニューラルネットワークを用いる場合，ユーザの興味を持つ状況を表す属性が出力として指定され，ルールの条件の候補とする他の属性が入力として指定される．

このグラフ構造は，初期設定で関連のない属性を連結しても，学習が進むにつれて，関連の深さによって連結の重みが調整される．このため，属性間の依存関係は連結の重みとして表現される．

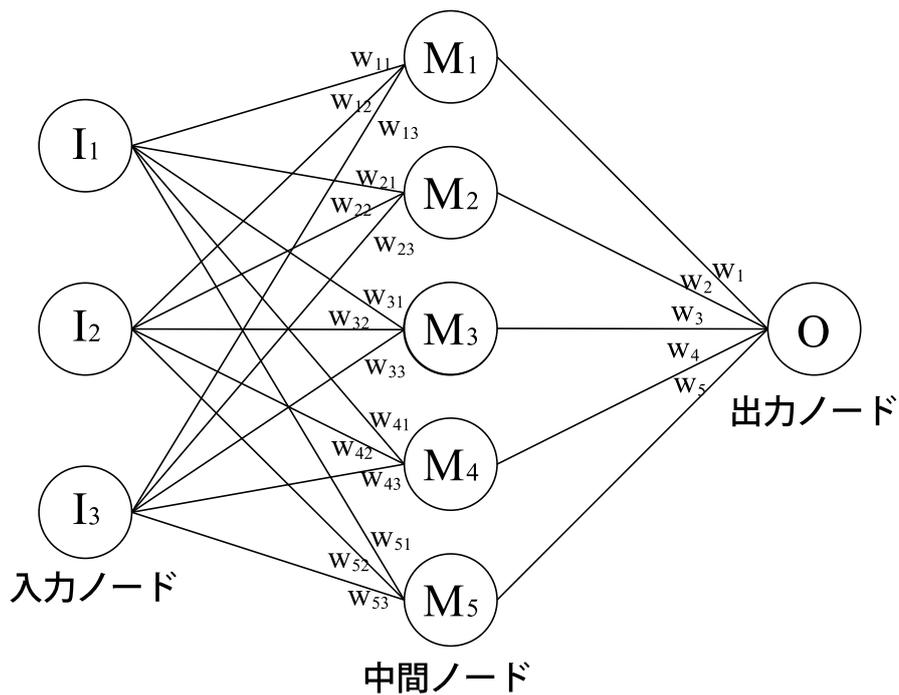


図 3.1: ニューラルネットワークの例

ニューラルネットワークで中間ノードを設定しない場合，入力ノードと出力ノードの連結の重みを参照すれば，依存関係が自動的に抽出できる．しかし，中間ノードを設定しなければ，連結の重みを w_i ，入力ノードから入力される値を I_i とすると， $\sum_{i=0}^n w_i I_i$ と表現でき，式 3.1 で示した統計的手法と同様の記述力であるため，これは現実的でない．

一方，一般的に用いられている形式である，中間ノードを設定したニューラルネットワークでは，記述能力が非線形判別関数まで上がるが，出力に対してどの中間ノードが影響を及ぼすかが分かるのみで，入力ノードからの影響を算出することが難しくなる．よって，条件の設定は出力として指定した属性に対して，入力として与えられた複数の属性が複雑に絡み合い，ある入力がどの値を取った場合に出力に反映されるかを解釈できない．このため，ルールとしての記述が難しくなる．

3.2.3 ベイジアンネットワーク

離散的な属性に対して複数の属性間の因果関係を条件付き確率で表したグラフ構造を構築する．ベイジアンネットワークの例を図 3.2 に示す．ベイジアンネットワークを用いる場合，初期設定として数値で表される属性を離散化し，全ての属性を入力した後，依存関係がある属性間にリンクを設定する．別のアルゴリズムの補助があれば，連続値をとる属性の離散化 [16]，属性間のリンク設定 [13] についても自動で行える．

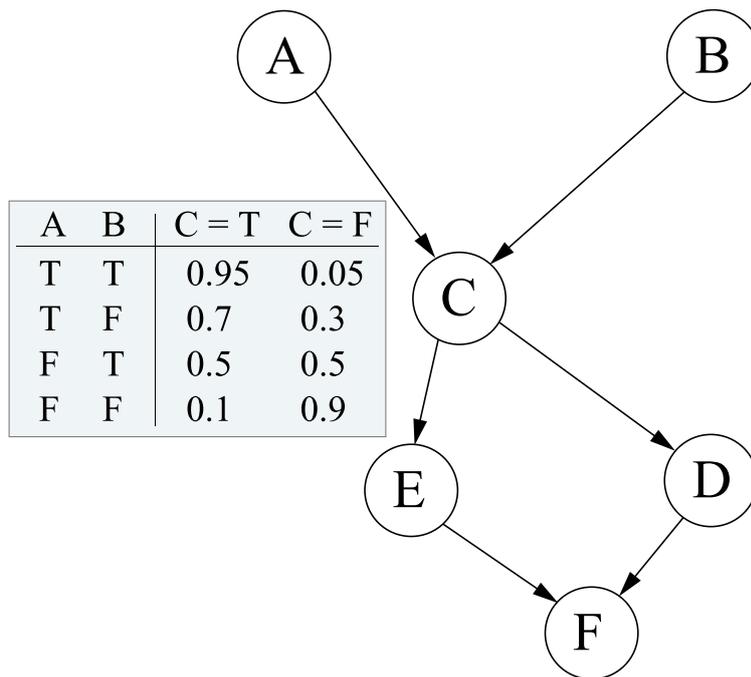


図 3.2: ベイジアンネットワークの例

このグラフは，ノードの連結が属性間の因果関係を示し，条件付き確率を参照しながら複数の連結をたどることで，条件として選択される属性が選別でき，条件に指定する値が設定できる．よって，ルールとの親和性も高い記述であるといえる．しかし，初期設定として行う必要がある，連続値の離散化や属性間のリンク設定がベイジアンネットワーク単体では利用できないため，情報依存関係抽出能力と条件設定能力が完全であるとはいえない．

3.2.4 決定木学習

与えられた属性とデータの集合，分類基準となるクラス属性を基に，トップダウンに木を構築していくアルゴリズムである．構築された木は決定木と呼ばれ，上からたどることで，フローチャートのように条件分岐を繰り返して分類結果に到達できる．決定木を用いる場合，ユーザが興味を持つ状況を表す属性をクラス属性に指定し，ルールの条件を候補とする他の属性を入力する．

作成された決定木は，分類基準に指定した属性を最も効率よく分類できる属性を，多数の候補となる属性から選出して構築されているため，条件として利用する属性を選出する要件は満たしている．また，決定木の構築アルゴリズムは当初，離散的な属性のみを対象としていたが，現在は連続値をとる属性に対して閾値を動的に設定し，離散化を行いながら構築するアルゴリズムも一般的となっている [33][34]．このため，全ての属性に対して条件に利用する値を設定する能力が備わっているといえる．ルール

との親和性に関しては，上に述べたように木の構造が条件分岐に該当するため，ルールとして記述するのは容易であるといえる．

3.3 決定木による普段の状況モデル

本研究では決定木学習を用いて，コンテキストの履歴を用いて普段の状況をモデル化する．そこで，木構造によって普段の状況がどのように記述されるかについて説明する．

まず，木構造について簡単に説明し，木構造で使われる用語について述べる．次に木構造の各部分が決定木ではどういった意味を持つかについて説明する．最後に，本研究で構築する決定木がどのように状況をモデル化するかについて述べる．

3.3.1 木構造

木構造とは，閉路を持たない 1 つの根を持つ連結有向グラフを指す．グラフとは，頂点の集合と，頂点同士のつながりを表す辺の集合である．連結グラフとは，任意に選んだ 2 つの頂点が，辺や他の頂点を介して，直接あるいは間接に結ばれているグラフである．有向グラフとは，辺が方向性を持つグラフである．

「根」とは，そこに入り込む辺を持たない頂点である．また，根以外の頂点はそこに入り込む辺を 1 つ持ち，出て行く辺を持たない頂点は「葉」と呼び，それ以外の頂点を「節」と呼ぶ．節と節，あるいは節と葉を結ぶ辺を「枝」と呼ぶ．これらの性質を持つ点の集合と辺の集合を合わせて「木」と呼ぶ [44]．図 3.3 に木構造の例を示す．

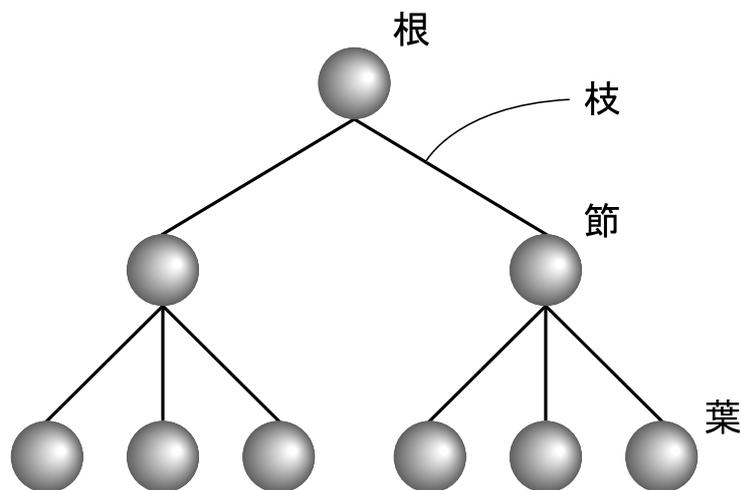


図 3.3: 木構造の例

3.3.2 決定木における木構造

決定木は、以上に述べた木の性質を持つ。それに加えて、決定木では根、節、葉や節などのそれぞれの部分にそれぞれ役割を持つ。根と節は、与えられたデータに対する質問文の役割を持ち、辺は、節における質問文の答えとなる選択枝の役割を持つ。また、根は一番最初の質問文となる。そして葉は、分類の終端であるクラスとなる。クラスとは、分類の基準となるデータ項目が取る値である。

決定木は通常、根を上にした形で描く。そして、上から順になぞりながら、データの質問文に答えていくことを繰り返すことで、分類の終端となるクラスへと到達する。決定木の例を図 3.4 に示す。

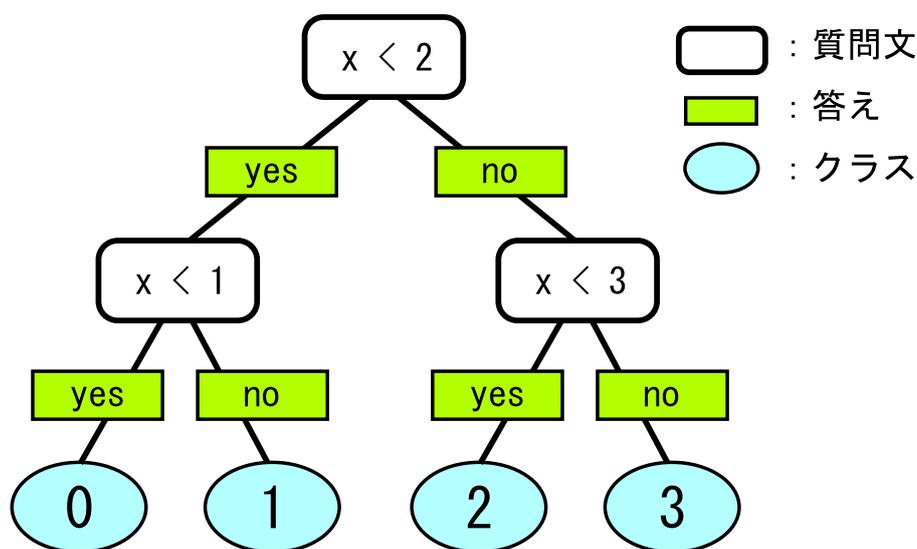


図 3.4: 決定木の例

この決定木は、0 ~ 3 の数当て問題の例である。0 ~ 3 の整数のいずれかを示すデータがあり、根にある質問文から始めて、該当する選択枝、次の質問... と繰り返すことにより、最終的に正解の数字へと到達する。この例で、もし $x = 2$ が与えられた場合、まず、根に該当する $x < 2$ の質問文に対して、答えが no であるため、右側の枝に進む。さらに $x < 3$ の質問文に対して答えることにより、実際の正解である 2 へと到達する。

3.3.3 普段の状況を表現する決定木

以下に本研究で行う、普段の状況のモデル化を、決定木の構造と対応させながら説明する。まずクラス属性として、ユーザが興味を持つ属性が割り当てられ、質問文としてそれ以外の属性を用いる。質問文に対する答えには、属性が取る値を割り当てる。これにより、上から辿ることでユーザが興味を持つ状況を識別できる決定木となる。図

3.5 に、本研究で用いる決定木の例を示し、表 3.2 に、木構造や決定木との対応表を示す。

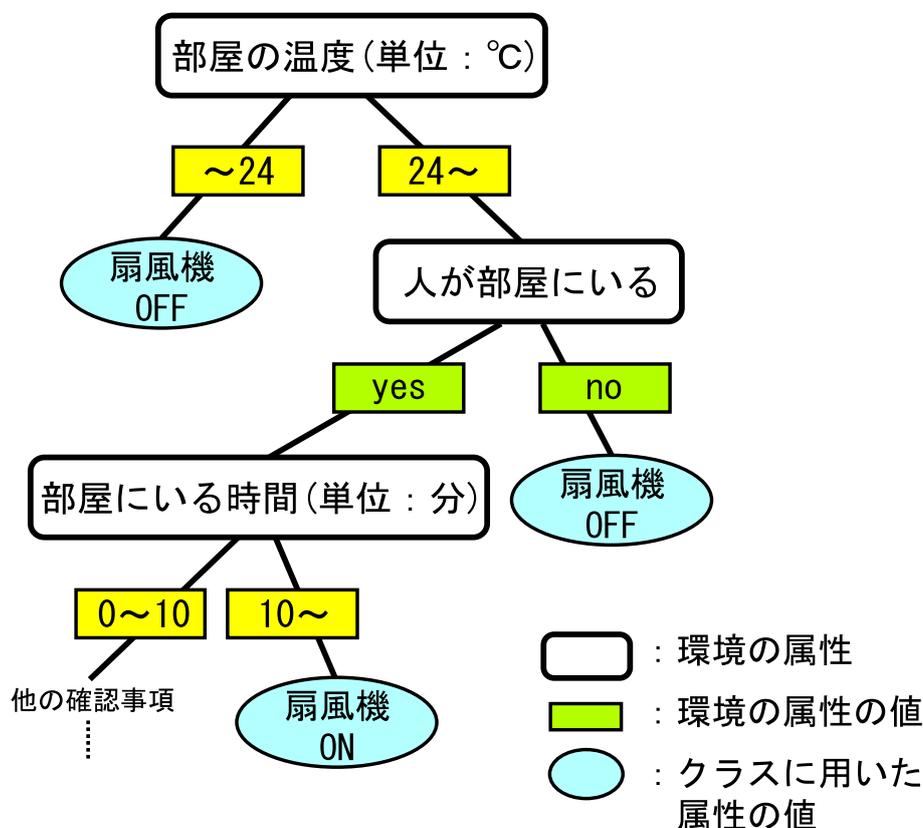


図 3.5: 扇風機の電源状態を興味ある属性に指定して作成した決定木

この例では、"扇風機の電源状態"という属性をクラス属性とし、その値が ON か OFF かをクラスとして、普段の状況を分類した木を表している。この木は、部屋の温度が24度以下の場合に扇風機が OFF の状態、部屋の温度が24度以上だが、人が部屋にいない場合に扇風機が OFF の状態、部屋の温度が24度以上で、人が部屋に10分いる場合には扇風機が ON の状態、を表現している。

3.4 決定木の構築手法と機能要件に対する妥当性

本節では、まず 3.3.3 で述べた本研究で用いる決定木の構築方法について説明し、次に 3.1 で挙げた機能要件をどのように満たすかについて述べ、ユーザが興味を持った状況がどのように条件として記述されるかについて述べる。

表 3.2: 木構造, 決定木, 定常状態を記述した木の対応表

木構造	決定木	定常状態を記述した木
根, 節	質問文	クラス属性以外の属性
枝	質問文の答え	上の節とした属性が取る値
葉	クラス	クラス属性

3.4.1 条件として利用する属性の選択基準

決定木は, クラス属性を効率良く分類するため, 節に選択する質問文として適切な属性を自動的に選択する. 本項では, この選択基準について説明し, 機能要件として挙げた, 条件に利用する属性として適した選択であることを示す.

決定木は, 節として設定する属性を選択する際, 分割の効率を示す基準として情報利得比を用いている. 情報利得比は, 情報利得を分割情報量で正規化したものである. 集合 X を属性 A によって分割したときの情報利得比 $GR_A(X)$ は, 情報利得を $G_A(X)$ とし, 分割情報量を $SI_A(X)$ とすると, 以下の式で表せる. 情報利得と分割情報量については後述する.

$$GR_A(X) = \frac{G_A(X)}{SI_A(X)} \quad (3.2)$$

情報利得は, シャノンの情報理論 [38] によるエントロピーが分割によってどれだけ減少したかを表す基準であり, この値が大きいほど, クラスによる分類が進んだことを表す. 集合 X のエントロピー $E(X)$ は, X が n 個の互いに重ならない集合に分割でき, X における i 番目の集合が占める割合を p_i とすると, $E(X) = -\sum_{i=1}^n p_i \log_2 p_i$ (bit) で表現できる. 学習対象の集合が全てクラスに分類された後は, p_i が 0 または 1 となり, エントロピーが 0 となるので, 分割前後のエントロピーの差分を表す情報利得も 0 となる. 上述の式での情報利得 $G_A(X)$ は分割後の集合を $S_j (j = 1, 2, \dots, k)$ として, 以下の式によって表される.

$$G_A(X) = E(X) - \sum_{j=1}^k \frac{|S_j|}{|X|} \times E(S_j) \quad (3.3)$$

分割情報量は, 選んだ属性が集合全体に対して, どれだけ細かく分割するかを表す基準であり, この値が大きいほど, 細かく分割することを意味する. 上述の情報利得のみで分割を行った場合, 分割が細かければ細かいほど, 情報利得の値は高くなる. そのため, 各要素を識別する ID による分割のように, 集合に含まれる他の要素に応用できない分割に対する評価も高くなる. そこで, 細かすぎる分割の評価を下げるために導入されたのが, この分割情報量である. 分割情報量は以下の式で表現できる.

$$SI_A(X) = - \sum_{i=1}^n \frac{|X_i|}{|X|} \log_2 \frac{|X_i|}{|X|} \quad (3.4)$$

この情報利得比を分類基準として用いることで、決定木を構築する際に、学習対象となる集合を効率良く分類し、かつ、学習対象となった集合以外にも応用が利く属性が節として選ばれる。

3.4.2 条件となる属性の値の選択

前項で述べた方法で節に入る属性を選んだ後、その節から出て行く枝を生成する。節に決定した属性が取る値の中で、学習対象となる集合内に含まれる値を枝とする。そして、生成された各々の枝に対して、学習対象となる集合から、節に入れた属性が枝に入れた値を取っているデータの集合を割り振る。ある枝に割り振ったデータの集合の中に、クラスがほぼ 1 種類しか含れず、他のクラスが紛れている量が誤差範囲と判断されたとき、その枝はクラスとなり、分類は終了する。

3.4.3 提案する条件の作成

決定木は、3.4.1 と 3.4.2 で述べたアルゴリズムを再帰的に行うことにより、木全体を構築する。

コンテキストウェアサービスに用いるルール条件として提案するコンテキストは、ユーザが興味を持った属性の値が結果として表現されている葉から、節と枝を根の方向に辿って行くことで得られる。

3.5 クラス属性の離散化

決定木は、クラス以外の属性が連続値を取る場合、構築時に自動的に離散化する手法が存在する。しかし、クラスに指定された属性が連続値をとる場合、事前に離散化する必要がある。本研究では、ユーザが興味を持つ属性が連続値を取る場合も考えられるため、本節でクラス属性に指定された属性が連続値を取る場合の離散化手法について述べる。

連続値を離散化するためには、連続値の分割点となる閾値を決定する必要がある。本研究では以下の点を考慮して離散化を行った。

- クラス属性の値が近く、それ以外の属性の値が近い場合は同じ離散値を取る
- クラス属性の適切な分割数が未知である

まず最初に挙げた点を考慮するために、近似性を算出する必要がある。近似性の算出方法については 3.5.1 で述べる。そしてクラス属性の値で並び替えを行い、隣り合

う情報が近いと判断できた順に統合していく．この方法はデンドログラムという統計処理の手法で実現できる [54]．デンドログラムは統合作業を任意の場所で止められるため，分割数を自由に決められる．これにより，適切な分割数が未知の場合でも柔軟に対応できる．図 3.6 にデンドログラムの例を示し，簡単に説明する．

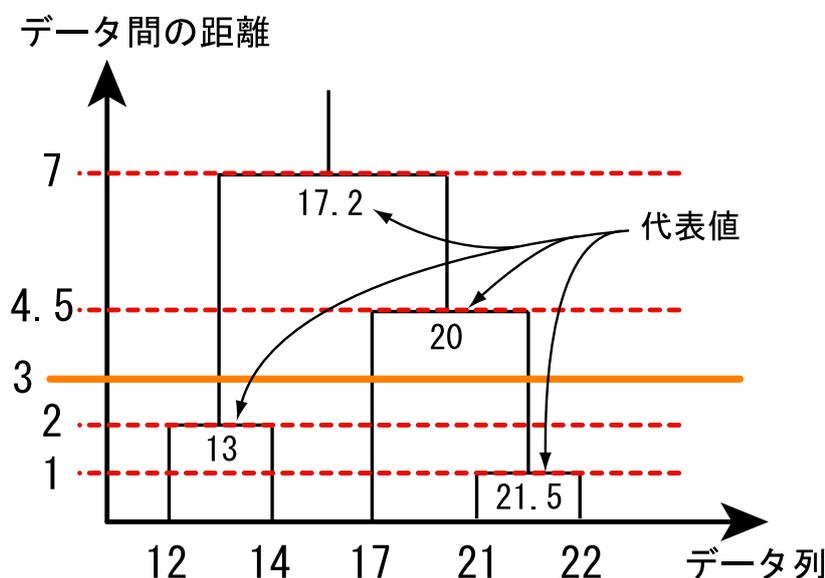


図 3.6: デンドログラムの例

デンドログラムは横軸がデータの並び，縦軸がデータの距離を表し，上に行くほどデータが統合されるグラフである．図の例ではデータを整数値とし，それぞれの数値の差を距離とした例を示している．各要素ごとの距離には差だけでなく，別の距離算出式を利用できる．距離の値が近いものから順に統合し，統合したデータ同士の距離が縦軸で表現される．統合したデータには新たに代表値を設定し，新たな要素として扱う．この作業を繰り返し行うことで，要素数を 1 つずつ減らしていく．従って，任意の要素数への離散化ができる．また図中の距離 3 で引いた線のように，全ての要素間の最低距離を指定した離散化ができる．

3.5.1 近似性の算出手法

クラス属性によって並び替えた隣り合う要素に対しての近似性を算出する．本研究では，3.2.1 で調べた数量化によってそれぞれの属性を数値で表現し，データ間の距離を用いて隣り合うデータの近似性を算出した．近似性の算出式としては，各要素を 1 軸とした n 次元の距離算出式を用いた．このとき要素間で距離を均等に評価するため，以下の処理を行った．

分布の調整

下で述べる正規化の前処理として、属性が連続値であった場合に分布の調整を行う。これは、センサから得られる値は普段ほとんど変化しないが、数値が変化するときには急激に変化するような、線形に変化していない値を補正する役割を果たす。例えば、振動センサや加速度センサが持ち運び可能なものについていた場合、置いてあるときは 0 に近い値でほとんどのデータは 0 を示しているが、持って歩いている時は変域が大きくなり、10 や 100 などに値が変化する。このとき 10 や 100 の差は実は少ないが、0 とは明確に区別したい、といった要求が考えられ、対数や二乗根を取る補正で要求に近づけることができる。また、人がある場所に滞在する時間は対数を取ると分析しやすいとの報告がある [19]。分布に適用する調整を汎用的に判別する方法として、線形回帰をすると仮定した際の値 \hat{y} からの標準偏差を s として、 $s = g(\hat{y})$ と表現したとき、補正関数は $h(y) = \int \frac{1}{g(y)} dy$ として与えられる [24]。本研究では、この値を参考に分布を調整する。

正規化

各要素の距離の最大値が 1 になるように、数値を調整する。属性が連続値であった場合は、最小値が 0、最大値が 1 となるように正規化、離散値であった場合は、 $\frac{1}{\sqrt{2}}$ を最大値に取る数量化を行った。例として、離散値の種類が A, B, C であるとき、 A は $(\frac{1}{\sqrt{2}}, 0, 0)$ 、 B は $(0, \frac{1}{\sqrt{2}}, 0)$ 、 C は $(0, 0, \frac{1}{\sqrt{2}})$ と 3 次元として表現され、離散値の距離はその空間内の距離となる。これにより、離散値が異なる値であれば距離は 1、同じ値であれば距離は 0 となる。

3.5.2 適切な閾値の設定

適切な閾値の設定とは、前述したデンドログラムを用いてどの程度までデータを統合するか、つまり終了条件を決定することを指す。本研究では、各要素間の距離に偏りがある場合、他の要素間に比べて距離が小さ過ぎる要素は統合するべきと判断する。また各要素間の距離に偏りがなくなった場合でも、離散化した後の要素数が多すぎてはユーザが把握しづらいため、記憶に留めやすい数として知られている 7 要素以下に抑える [29]。

3.6 本章のまとめ

本章ではコンテキストウェアサービスに用いるルールの条件を、自動的に抽出するための要件について考え、利用するアルゴリズムについて考察した。また、本研究での工夫点について述べた。次章では本研究で作成するソフトウェア、QuPen の設計について述べる。

第4章 QuPenの設計

本章では，QuPenの設計について述べる．まず全体の構成と動作概要について述べ，次に本システムを構成する各モジュールについて述べる．

4.1 QuPen の概要

本節では，QuPen の設計概要として，ハードウェア構成，ソフトウェア構成と全体の動作概要を述べる．

4.1.1 ハードウェア構成

本節では，システムのハードウェア構成について述べる．まず，本システムの動作環境として想定している環境について述べ，次に本システムの構成要素について説明する．

システム想定環境

本システムは，家やビルなどの建物内に 1 台以上のサーバが存在し，2.2.2 で述べたようにセンサや機器がネットワークに接続され，情報取得や操作ができる環境上で動作する．また，本研究が対象とするルールに基づいたコンテキストウェアサービスを提供するシステムも同一ネットワーク上にあり，ネットワーク上に存在するセンサや機器は本システムと同様にアクセスできる．センサや機器の名前は，ネットワーク上から取得でき，両システムで統一されているものとする．

構成要素

本システムが動作するサーバ，情報を取得するセンサ，機器について以下に説明する．また，構成要素がどのように配置されているかを，ハードウェア構成図として図 4.1 に示す．

- サーバ

部屋単位，あるいは建物単位で設置された，PC やワークステーションなどの端末である．QuPen はその端末上で動作する．その端末は QuPen 専用ではなく，ホームサーバとして別のシステムが動作していても良い．

- センサ

位置情報，温度，湿度や音量などの情報を環境から取得するデバイスである．環境に設置される形やユーザが携帯する端末に付随する形で存在している．有線あるいは無線でネットワークに接続されており，ネットワークを通して情報を取得できる．

- 機器

照明，AV 機器やエアコンなどのデバイスで，環境に設置されているか，ユーザに携帯されている．センサ同様，有線あるいは無線でネットワークに接続されて

おり，電源の ON/OFF や動作設定など動作状態を表す情報がネットワーク上から取得できる．

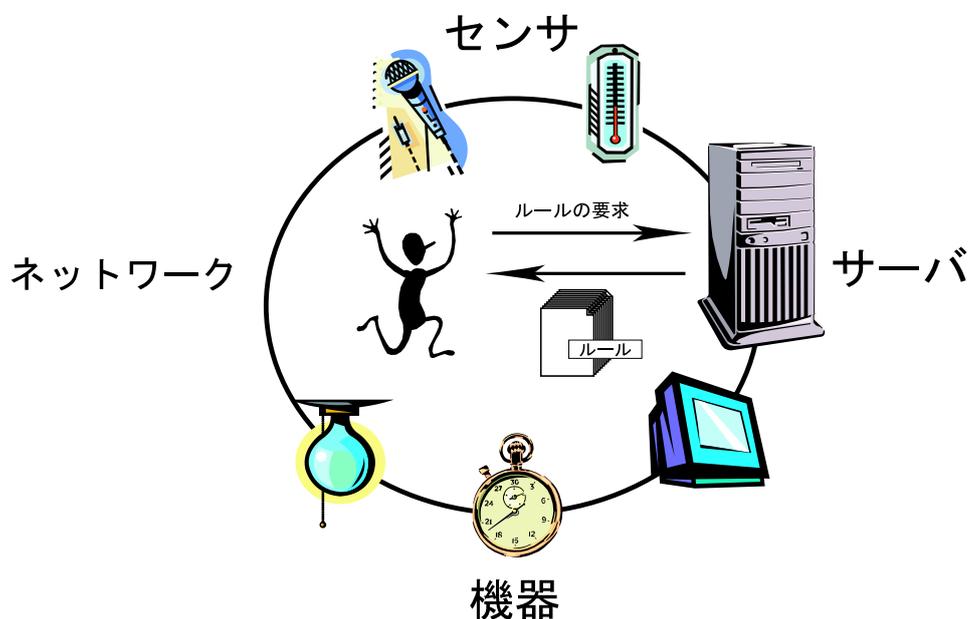


図 4.1: ハードウェア構成図

4.1.2 ソフトウェア構成

本システムは，大きく分けて 5 つのモジュールから構成されている．センサや機器から情報を集める情報取得部，集めた情報を履歴として保存する履歴管理部，ユーザからの要求を受けて履歴から決定木を作成する際に，前処理として 3.5 で述べた処理をし，履歴のフォーマットを行う統計処理部，指定された属性を元に普段の状況を表現する決定木を作成する木作成部，作成された木を元にユーザに提案する条件を出力する条件出力部が存在する．各部の詳細については，4.2 で述べる．本システムのソフトウェア構成図を図 4.2 に示す．

4.1.3 動作概要

本システム全体の動作概要について述べる．本システムの動作は情報蓄積作業と条件作成作業の 2 つに分かれる．それぞれの動作手順について以下に説明する．

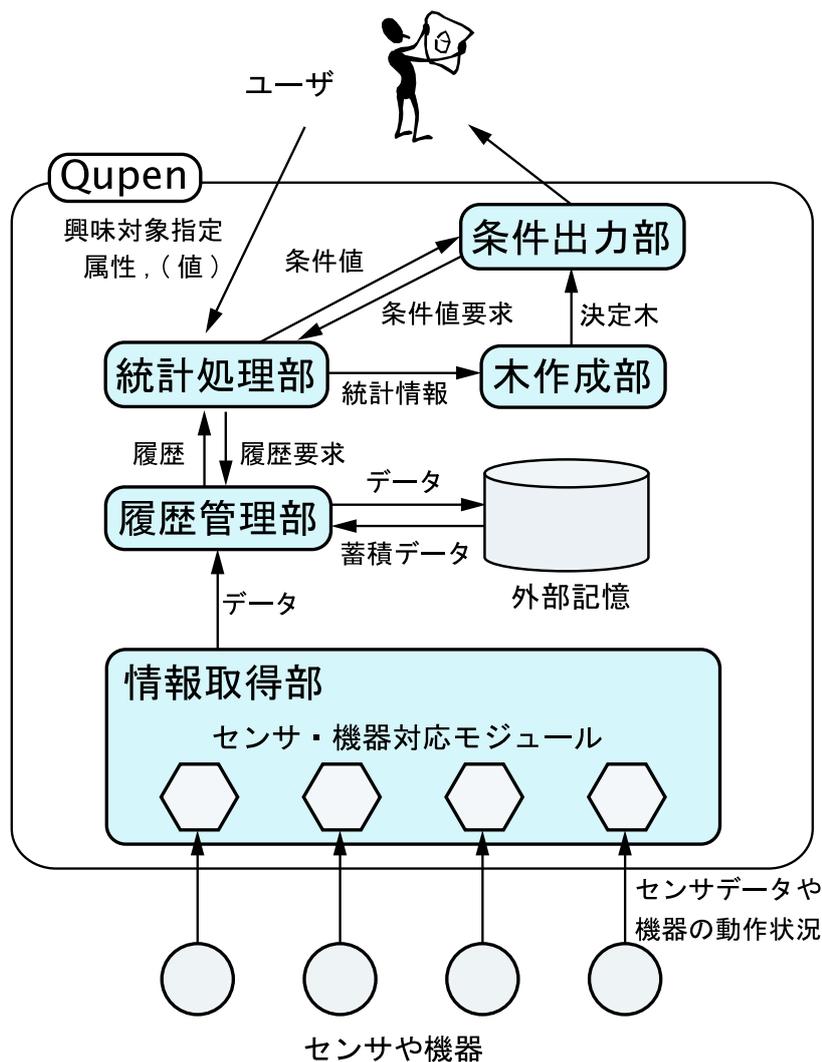


図 4.2: ソフトウェア構成図

情報蓄積作業

情報蓄積作業のシーケンス図を図 4.3 に示す。この作業はシステム起動時から継続して行われる作業であり、情報取得部と履歴管理部が動作する。ネットワーク上にあるそれぞれのセンサや機器に対応する情報取得部内のモジュールが情報を取得し、履歴管理部にデータが送られる。履歴管理部に送られた情報は、履歴として蓄積される。

条件作成作業

条件作成作業のシーケンス図を図 4.4 に示す。この作業はユーザからの要求によって行われる作業であり、主に統計処理部、木作成部と条件出力部が動作する。ユーザは、興味を持つ情報を表す属性、必要であればその属性の値を本システムに対して入

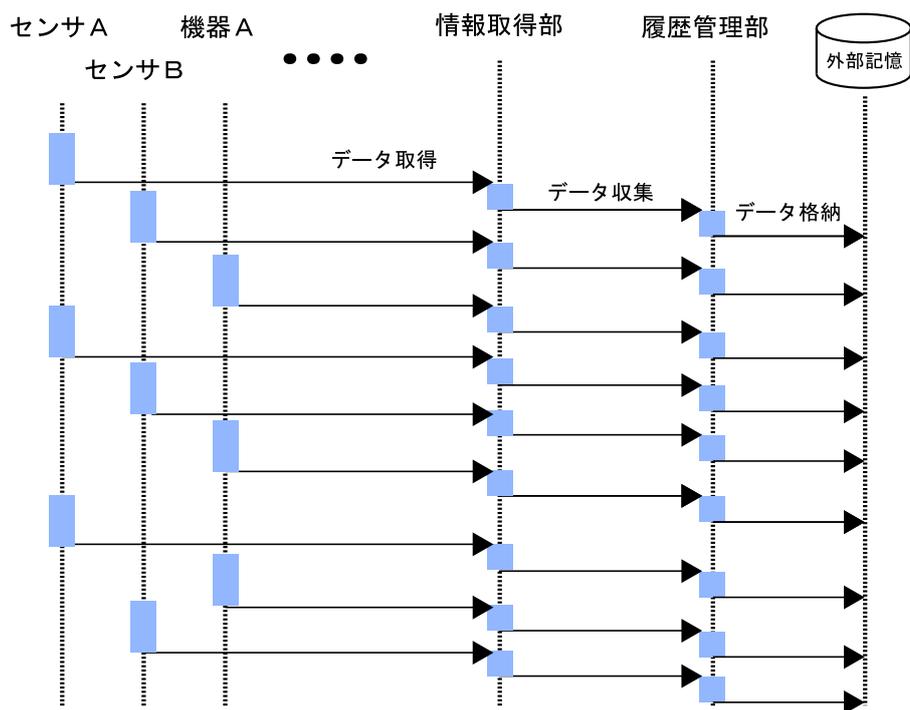


図 4.3: 情報蓄積作業シーケンス図

力する。その属性が連続値をとる場合，統計処理部が離散化を行う。こうして処理された離散的な値をクラス属性として木作成部が普段の状況をモデル化した決定木の作成を行う。条件出力部は木作成部が記述した決定木を基に，ユーザに対して条件を提案する。

4.2 各モジュールの設計

本節では，QuPen を構成する各モジュールの設計について述べる。それぞれのモジュールに対して，入力される情報，内部で行う処理，出力される情報について詳細に述べる。

4.2.1 情報取得部

本モジュールはセンサや機器からネットワークを介して情報を取得し，履歴管理部に渡す。本システムが扱うセンサおよび機器のそれぞれに対応したセンサモジュールおよび機器状態管理モジュールから構成される。

- 入力情報

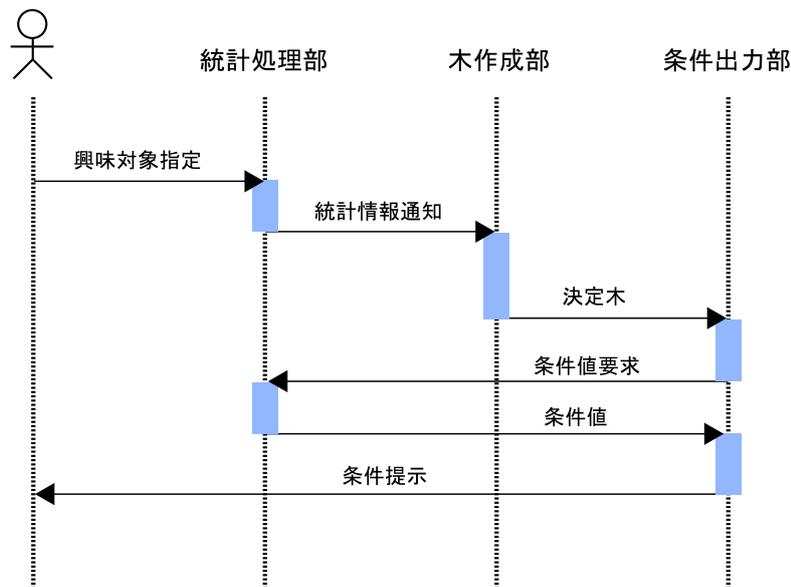


図 4.4: 条件作成作業シーケンス図

入力情報は，センサや機器からの raw 情報である．raw 情報とは，センサや機器から直接取得する情報であり，ヘッダやフッタなど本システムが必要とする属性以外の情報を含む場合がある．

- 内部処理

入力された情報から，不要な情報の切り捨てや形式の変換などのそれぞれのセンサや機器に合わせた解釈をする．また，それぞれの情報がどのセンサや機器属性から取得したかを識別するために，属性の識別情報を付加する．そして，全ての情報を 3.2 で述べた離散値と連続値として系統的に区別できる形式として表現する．

- 出力情報

出力情報は属性情報が付加された値であり，離散値または連続値のいずれかである．

4.2.2 履歴管理部

本モジュールは情報取得部から取得した情報を履歴として蓄積する．情報取得部を構成しているセンサや機器を扱うモジュールそれぞれから更新情報を受け取る情報収集モジュールと，履歴情報に更新情報を付加し，統計処理部が受け取る形として保存する履歴蓄積モジュールから構成される．

- 入力情報

情報取得部でそれぞれのセンサや機器を扱うモジュールから，属性情報が付加された離散値あるいは連続値が与えられる．また，統計処理部からの要求を受けつける．

- 内部処理

受け取った値を属性ごとに分類し，値が入力された時間を記録する．そしてその情報を履歴情報に新しく付加する．

- 出力情報

統計処理部に，列ごとに別の属性に分けられ，行がある瞬間のコンテキストを表現する 2 次元の配列情報を渡す．

4.2.3 統計処理部

本モジュールはユーザからの要求を受けて，決定木を作成する作業を開始する役割を果たす．ユーザからの要求には，クラスとする属性，必要に応じてその属性が取る値が含まれる．要求情報を管理するモジュールと，離散化を行うモジュールで構成される．クラスとする属性が連続値だった場合，履歴情報を離散化して木作成部に渡す．

- 入力情報

ユーザから，決定木のクラス属性として扱う属性，必要に応じてその属性がとる値を指定され，決定木の構築要求を受ける．その要求を受けた後，履歴管理部から渡される履歴情報を読み込む．

- 内部処理

クラス属性が連続値であった場合，3.5 で述べた離散化を履歴情報に対して行い，木作成部に渡す形式へと形式変換する．クラス属性が離散値であった場合は，履歴の形式を変換作業のみを行う．

- 出力情報

クラス属性が離散値を取る履歴情報を木作成部に渡す．ユーザからの要求にクラス属性の値が含まれていた場合，条件作成部からの要求を受けて，その情報を渡す．

4.2.4 木作成部

本モジュールは履歴情報からユーザから指定された属性をクラス属性とした普段の状況を表す決定木を構築する．クラス属性以外の属性が連続値であった場合，自動的に離散化できる決定木構築アルゴリズム [34] を用いる．

- 入力情報
統計処理部から，クラス属性が離散値である履歴情報が渡される．
- 内部処理
本モジュールは 3.4 で述べた手法を用いて，決定木を構築する．作成された決定木のクラスに反例が含まれる場合もあるため，確信度を記述する．
- 出力情報
本モジュールで構築した普段の状況を表す決定木が条件出力部に渡される．

4.2.5 条件出力部

本モジュールは木作成部で構築された決定木を用いて，ユーザが興味を持つ状況の条件を抽出する．本モジュールはユーザが指定した属性に対しての条件の提案を行う．

- 入力情報
木作成部で構築された普段の状況を表す決定木を受け取る．ユーザが要求にクラス属性の値を指定した場合は，クラス属性の取る値が入力される．
- 内部処理
決定木を下からたどり，クラス属性が取る値ごとに条件を算出する．ユーザからの要求にクラス属性の値が指定されていた場合は，その値を取る場合の条件のみに絞る．
- 出力情報
ユーザへ，本システムが提案する条件が出力される．各条件には木作成部で算出された確信度が付加される．

4.3 本章のまとめ

本章では，QuPen の設計について，システムを構成するモジュールとその動作，処理内容とモジュール間でやり取りされる情報について述べた．次章では，QuPen の実装について述べる．

第5章 QuPenの実装

本章では，前章で述べた設計を元にした QuPen の実装について述べる．まず実装概要について述べ，次に各モジュールの実装について述べる．

5.1 実装概要

QuPen は実装言語として主に Java を用い、既存の実装として C , Python を利用し、Java の `java.Lang.Runtime` を用いて外部プログラムとして実行した。開発および実行環境はそれぞれ、J2SE 1.5.0_06-b05, gcc 3.4.4, Python 2.4.1 を用いた。

本実装では、情報取得部と履歴管理部、統計処理部、木作成部、条件出力部に分けてそれぞれパッケージ化した。次節で各モジュールの詳細について述べる。

5.2 各モジュールの実装

本節では、QuPen を構成する各モジュールの実装について述べる。各モジュールでの使用言語、既存のソフトウェアを利用した場合はその引用元を述べ、Java を用いて実装した部分についてはクラス構成と主要なクラスについて述べる。

5.2.1 情報取得部

情報取得部は Java を用いて実装した。RS-232 通信が必要なセンサや機器は Java Communications API を用いて実装した。情報取得部では、本システムがコンテキストとして扱う情報を抽象化するために、Data クラス、2.2 に挙げたコンテキストとして扱う情報の提供元を抽象化するために、DataInput インターフェースを実装した。Data クラスのサブクラスとして、離散値として DiscreteData クラス、連続値として ContinuousData クラスを実装した。また DataInput のサブインターフェースとして、2.2.1 で述べた人が入力した情報を扱う PersonalInput インターフェース、2.2.2 と 2.2.3 で挙げた、センサからの情報を扱う SensorInput インターフェース、2.2.4 で挙げた、時間を情報源とした TimeInput インターフェースを用意した。この抽象化により、システムで扱う全てのデータは DataInput から得られる Data として統一的にアクセスできる。本モジュールのクラス図を図 5.1

Data クラスはインスタンス変数として、情報の属性と値を表す `java.lang.String` 型の `attribute` と `java.lang.Object` 型の `data` を持ち、それらはコンストラクタによって初期化される。しかし、このコンストラクタは外部パッケージからは参照できないため、実際の処理は Data クラスを継承した、DiscreteData クラスまたは ContinuousData クラスを介して呼ばれ、`data` には `java.lang.String` 型または `java.lang.Double` 型が代入される。

DataInput インターフェースでは、属性のリストを `java.lang.String` の配列として返すメソッド `getAttributes()` と、属性を指定して Data オブジェクトを取得するメソッド `Data getData(java.lang.String attribute)` を定義した。また PersonalInput インターフェースでは、誰に所属する情報であるかを返す `java.lang.String getPerson()` メソッドを定義し、SensorInput インターフェースでは、最新データに更新された時間

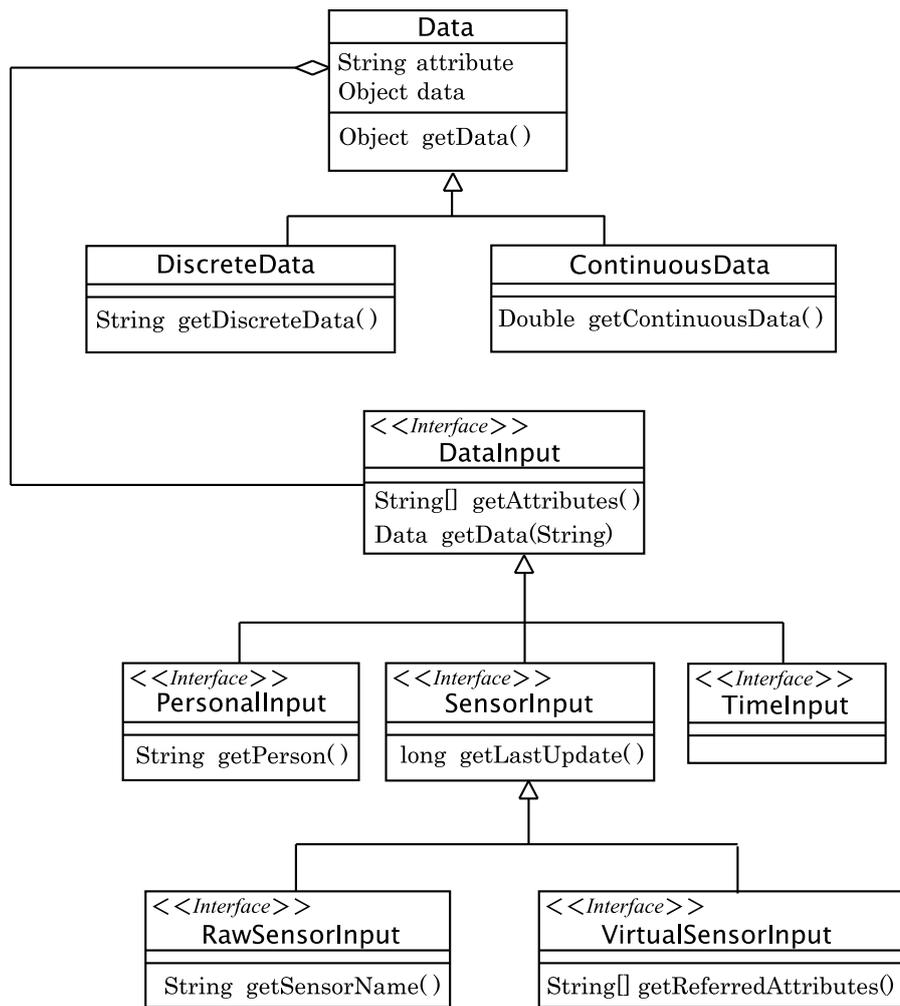


図 5.1: 情報取得部クラス図

を `java.lang.Date` の `getTime()` メソッドから得られる値で返す `long getLastUpdate()` を定義した。

本研究で実際に実装したのは `SensorInput` インターフェースであり、さらにサブインターフェースとして `RawSensorInput` インターフェースと `VirtualSensorInput` インターフェースを用意して 2.2.2 と 2.2.3 で述べた、センサから直接取得した値と加工した値を区別した。`RawSensorInput` インターフェースでは、`java.lang.String` 型を返す `getSensorName()` メソッドを定義し、もし同じ属性を返す異なる種類のセンサがあり、補正をかけたい場合に対応した。`VirtualSensorInput` インターフェースでは加工元となった属性を取得できる、`java.lang.String` 型の配列を返す `getReferredAttributes()` メソッドを定義した。

5.2.2 履歴管理部

履歴管理部は Java を用いて実装した。本モジュールは `DataInputManager` クラスを中心に動作する。`DataInputManager` には、`DataInput` の登録を行う `addDataInput(DataInput input)` メソッドが用意されており、システム動作開始時にセンサや機器の登録が行われる。`DataInputManager` は登録された `DataInput` クラスの属性を `getAttribute()` メソッドによって取得し、`getData(java.lang.String attribute)` メソッドによって `Data` オブジェクトを取得する。情報取得部の `DataInput` インターフェースを実装したクラスから各属性のデータを `Data` クラスとして取得し、2次元の `String` 配列として統計処理部に渡す。このデータを CSV ファイルとして記述したサンプル例を図 5.2 に示す。1 行目が属性の記述で、それ以降はデータを表す。属性が連続値をとる場合、属性の名前の最後に `(cont)` と付加することで、識別した。

```
RF_sofa, RF_door, RF_desk, RF_room, TV, light, temp(cont), stay(cont)
FALSE, FALSE, FALSE, FALSE, OFF, OFF, 15, 0
FALSE, FALSE, FALSE, FALSE, OFF, OFF, 14, 0
FALSE, FALSE, FALSE, FALSE, OFF, OFF, 15, 0
FALSE, TRUE, FALSE, TRUE, OFF, OFF, 15, 3610
TRUE, FALSE, FALSE, TRUE, ON, ON, 15, 3610
TRUE, FALSE, FALSE, TRUE, ON, ON, 17, 3610
TRUE, FALSE, FALSE, TRUE, ON, ON, 19, 3610
FALSE, TRUE, FALSE, TRUE, OFF, ON, 20, 3610
FALSE, TRUE, FALSE, TRUE, OFF, ON, 19, 3610
FALSE, TRUE, FALSE, TRUE, OFF, OFF, 19, 3610
FALSE, TRUE, FALSE, TRUE, OFF, OFF, 18, 3610
FALSE, TRUE, FALSE, TRUE, OFF, OFF, 15, 127
FALSE, TRUE, FALSE, TRUE, OFF, ON, 15, 127
FALSE, FALSE, TRUE, TRUE, OFF, ON, 15, 127
FALSE, FALSE, TRUE, TRUE, OFF, ON, 15, 127
FALSE, TRUE, FALSE, TRUE, OFF, ON, 15, 127
FALSE, TRUE, FALSE, TRUE, OFF, ON, 15, 127
FALSE, TRUE, FALSE, TRUE, OFF, OFF, 15, 127
```

図 5.2: 履歴管理部の出力データサンプル

この例で扱っているデータについて、簡単に説明する。左から 4 列は人の滞在状態を指し、それぞれ、ソファ付近、ドア付近、机付近、部屋全体を表している。また、部屋全体は 2.2.3 で述べた、他の 3 つの状態を統合した値であり、いずれかが `TRUE` だった場合に `TRUE` となっている。左から 5 列目はテレビの電源状態を表し、6 列目は部屋の電灯の状態を表している。そして 7 列目は温度センサから取得した温度、8 列目は 1 度部屋に入ってから出るまで滞在した時間を滞在していたデータ全てに付加した属性を表している。

5.2.3 統計処理部

統計処理部では、履歴管理部から渡された情報をまとめて管理する `SampleDataMap` クラスを中心としてデータを扱う。`SampleDataMap` は `java.util.HashMap<java.lang.String, java.util.Vector>` を継承したクラスで、属性をキーにしてデータが保存される。ユーザの要求に指定された興味対象の属性を決定木のクラス属性として、図 5.2 に示した列の情報を選択し、後述する C4.5 決定木構築プログラムに読み込ませる形式に変換する。

クラス属性が連続値だった場合、3.5 で述べたデンドログラムを用いて離散化する。`SampleDataMap` が保持するそれぞれの属性ごとに、データの集合に対して、数値化、分布調整、正規化などの処理を適用する必要があるが、これらを `SampleFilter` インターフェースとして抽象化した。`SampleFilter` インターフェースでは `java.util.Vector` 型のオブジェクトを返す `apply(java.util.Vector)` メソッドが定義され、数値化、分布調整、正規化などの処理を統一的に扱える。デンドログラムで用いる計算に必要となる、要素間の距離算出と統合した際の代表値算出に用いるアルゴリズムも、それぞれ `DistanceCalculator` インターフェースと `RepresentativeSelector` インターフェースとして抽象化し、異なるアルゴリズムを統一的に扱った。

本モジュールの最後の処理として、C4.5 決定木構築プログラムに渡すための形式変換について述べる。C4.5 が読み込むファイルは属性ファイルとデータファイルの 2 種類である。データファイルは図 5.2 で示した情報から、一番上の行の属性情報と、クラス属性に指定された列を削除したデータである。属性ファイルの形式の例として図 5.3 に、先ほどのデータを用いて滞在時間をクラスに指定した場合、生成される属性ファイルを示す。

```
0, 3610, 127.  
  
RF_sofa: FALSE, TRUE.  
RF_door: FALSE, TRUE.  
RF_desk: FALSE, TRUE.  
RF_room: FALSE, TRUE.  
TV:      OFF,   ON.  
light:   OFF,   ON.  
temp:    continuous.
```

図 5.3: 属性ファイル例

このファイルはクラスが取る値が最初に示され、次にそれぞれの属性について離散値の場合は取り得る値を、連続値の場合は `continuous` というタグを指定する。データファイルは属性ファイルで定義された順に、左から列ごとに属性のデータを表現する。この例では連続値であったクラス属性が 3 種類の値しか取らなかったため、それらの

値を離散値として扱っている。

5.2.4 木作成部

木作成部は Java, C, Python が用いられている。決定木の実装として、既存の C4.5 アルゴリズムの実装を利用した [17]。木作成部では、統計処理部から得られた C4.5 の属性定義ファイルとデータファイルを元に C 言語で書かれた C4.5 プログラムファイルを実行し、結果として得られた決定木を Python で書かれたコードを用いて XML で表現する。Java から両プログラムを実行するために、java.lang.Runtime クラスの exec(java.lang.String command) を用い、戻り値として得られる java.lang.Process クラスを用いて標準出力への出力結果を取得した。本モジュールから出力されるデータ例として、先ほどのデータを元に作成した木を図 5.4 に示す。

```
<question term="RF_room">
  <answer term="FALSE" operator="equal to">
    <class>0</class>
    <accumulate>100.00</accumulate>
  </answer>
  <answer term="TRUE" operator="equal to">
    <question term="temp">
      <answer term="15" operator="less or equal than">
        <class>127</class>
        <accumulate>81.82</accumulate>
      </answer>
      <answer term="15" operator="more">
        <class>7200</class>
        <accumulate>100.00</accumulate>
      </answer>
    </question>
  </answer>
</question>
```

図 5.4: 決定木を表現する XML の出力結果例

決定木を表現する XML は、質問文を表す question 要素と、答えを表す answer 要素、分類結果である class 要素と分類結果に正確さを表す accumulate 要素から構成される。question 要素は分類の基準として用いられる属性を含み、answer 要素は条件分岐を表す値を term として含む。answer 要素はさらに演算子として operator を含む。question 要素、answer 要素では含まれた要素は木構造の下位ノードを指し、class 要素では含まれた文字列は分類結果を表す。accumulate 要素では含まれた値は 正例 の値を示し、分類結果の正確さを表現する。
正例 + 反例

この例で表現している滞在時間に対する分類結果を，上から順に説明する．部屋に人がいなかった場合，滞在時間は 0 秒で，部屋に人がいた場合，温度属性でさらに分岐する．温度が 15 度以下の場合，滞在時間は 127 秒と分類され，15 度以上のとき，7200 秒と分類された．0 秒，127 秒，7200 秒に分類された確信度はそれぞれ，100, 81.82, 100 である．

5.2.5 条件出力部

条件出力部では，2.1 で述べたルールの定義を参考に Rule クラスを定義した．Rule クラスは条件を表すクラスである Condition クラスを複数持ち，結果を表す Action クラスを 1 つ持つ．また，Rule クラスは double の確信度を表すインスタンス変数を 1 つ持つ．Rule クラスは ”if 条件 then 結果” として出力する java.lang.String 型を返すメソッド，getIfThenString() を持つ．

5.3 本章のまとめ

本章では，QuPen の実装について，システムを構成するモジュールの実装，やりとりされるデータについて述べた．次章では，QuPen の動作実験について述べる．

第6章 評価

本章では，QuPenの実験と評価について述べる．まず実験について説明し，その結果を示す．また，定性評価として関連研究との比較を行った．

6.1 実験

本節では，QuPenの動作実験について述べる．まず実験の概要について述べ，実験環境やシナリオについて解説し，実験結果について述べる．

6.1.1 実験概要

センサや機器を用いてコンテキストが取得できる環境において，簡単なシナリオを用意し，ルールとして利用可能な条件が作成できるか検証するため，QuPenの動作実験を行った．また取得したコンテキストの中から，条件作成に利用するデータを増減させ，結果にどう反映されるかを試した．

6.1.2 実験環境

実験環境として，慶應義塾大学湘南藤沢キャンパス 棟 208 号室を利用した．センサや機器をネットワークに接続し，それらの情報をネットワークから取得できる環境を構築した．今回利用したデバイスを以下に挙げ，図 6.1 に部屋の見取り及び機器の配置を示す．

センサ

- RF Code Spider
RF Code 社が提供する 308.8MHz 帯の赤外線電波を発するアクティブタグを認識できる，RF-ID 認識デバイスである [40]．今回は環境側にタグ認識センサを設置し，タグを人が所持し，人がいるかいないかを識別するために用いた．単体ではネットワーク機能を持たないため RS-232C で PC に連結し，ネットワーク上にデータを転送した．
- uPart
uPart[10] はドイツの TecO 研究所 [4] によって開発された，極小センサノードであり，振動，温度，照度の 3 つの情報が取得できる．それぞれのセンサノードは 315MHz 帯の無線通信機能を持ち，取得したデータをセンサネットワーク上に送信する．センサネットワーク上に流れたデータは，ブリッジを介して IP ネットワークに転送され，PC でデータを取得できる．

機器

- テレビ
ネットワーク制御ができないテレビを，ネットワーク対応電源制御装置 [11] に取り付けて，ネットワークから情報取得可能な機器として用いた．テレビのスイッチを使わずに，電源制御プログラムを介して電源制御を行うことで，テレビの ON, OFF が取得できる．

- コーヒーメーカー
テレビと同様ネットワーク対応電源制御装置を用いて，電源情報を取得した．



図 6.1: 実験環境の見取り図

6.1.3 実験内容

本論文における実験は，実機を用いてどのようなルールが作成できるか試すことを目的として行った．ユーザは以下に述べる形で機器を配置し，実験シナリオに述べる作業を 5 回ほど繰り返した．

センサ及び機器の配置

- センサ

RF Code Spider は実験環境を全てカバーするように設置した．また，uPart は椅子，PC，机に取り付けて，それぞれ椅子に人が座ってるかどうか，PC を利用しているかどうか，机にあたる光量から部屋の電灯の状態を判別するために利用した．

- 機器

図 6.1 に示したように配置した．それぞれの機器の操作はユーザが座った場所から行える．

実験シナリオ

ユーザが部屋に入ってきて、部屋の電気をつけ、椅子に座る。次にテレビの電源を入れ、しばらく見た後、コーヒーマーカを起動する。テレビを消し、PCを開く。しばらくするとPCを閉じ、部屋を出て行く。

6.1.4 実験結果

実験の結果、コーヒーマーカの電源状態をクラスとして、作成された決定木を図 6.2 に示し、それを元に記述したルールを図 6.3 示す。

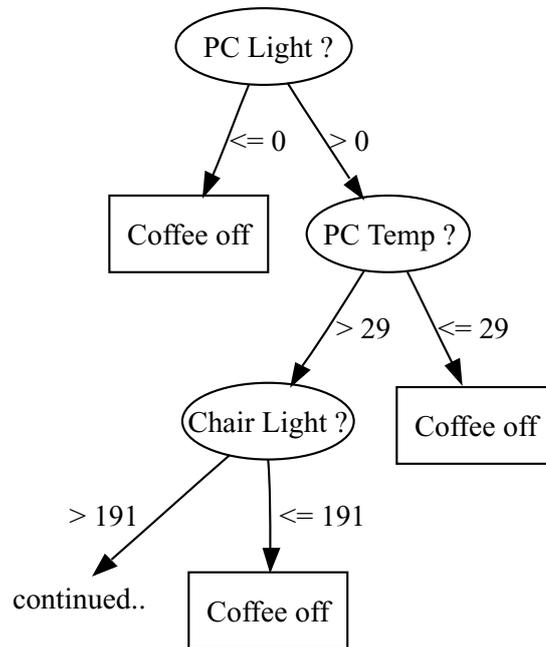


図 6.2: 実験結果として得られた決定木

6.2 定性評価

本節では、定性評価として、実験結果の分析、関連研究との比較を行う。

6.2.1 ルールの記述

今回記述できた決定木は、PC が開かれていない場合はコーヒーマーカは動作していない、PC が開かれていても温度が高くなければコーヒーマーカは作動していない、その 2 つの条件を満たした場合にも、人が椅子に座っておらず、光が直接当たってい

```
if PC_Light <= 0 : then
    Coffee off;

if PC_Light > 0 : then
    PC_Temp <= 29 : then
        Coffee off;

if PC_Light > 0 : then
    PC_Temp > 29 : then
        Chair_Light <= 191 : then
            Coffee off;
```

図 6.3: 実験結果として得られたルール

ればコーヒーメーカーは作動していない，という条件を表している．さらに細かい条件も作成されたが，省略した．

作成した条件を，コンテキストウェアサービスに用いるためには，上述したようにセンサの値を解釈して適切な条件かどうか判断する必要がある．しかし，照度のような閾値の設定が困難な値を自動的に設定することで，条件作成の手間は減ると考えられる．

6.2.2 関連研究との比較

本項では，本システムの機能を関連研究と比較する．比較対象として，履歴からサービスを自動化するシステムである SmartMacro[32] と Synapse[49] を挙げる．

SmartMacro

SmartMacro は，コンテキストの履歴から，サービスの連続を見て次のサービスを決めるため，2.3.1 で述べた日々利用しているサービスのうち，同じコンテキストで同じパターンで利用しているサービスについては，一つのサービスから連続して自動化できる．本研究では，サービスごとに問い合わせる必要があるため，この種類のサービスを自動化するには SmartMacro が優れているといえる．このようなサービスの例としては，朝起きてから外出するまでのサービスなどが考えられる．

しかし，SmartMacro では，同じサービスを繰り返すとは限らない例では，正確な予測ができない．ふと思立って利用するようなサービスや，休日を過ごす場合など時間的制約が無くパターンが存在しない状況には向いていない．本研究で用いた決定木では，多くコンテキストの中から，サービスを利用することが判別できる属性を自動的に抽出してくれるため，コンテキストの多くが同じでなくとも，サービスを利用

する状況の条件を抽出できる。これは、センサや機器が多くなればなるほど、コンテキストの多くが同じになるのは難しくなると考えられるため、このようなサービスでは本システムの方が優れているといえる。

Synapse

Synapse は、イベントの連続を見ているため、SmartMacro 同様に同じパターンで利用しているサービスについては予測できる。また、Synapse には、提案機能があるため、ユーザが明示的に指示を出さずとも作業の自動化が推薦される。この部分は、上述した SmartMacro だと、ユーザが適切なタイミングで指示を出さないと、サービスの連続が自動化できないため、Synapse がより優れているといえる。ちなみに本研究では、ユーザが指定する状況は任意のタイミングであるため、タイミングの問題は無い。

しかし、この指示を出さずともルール化を行い提案することは、場合によってはユーザの邪魔になってしまう。現在、この問題に対して、禁止ルールを記述する方法が提案されているが、禁止ルールはどうしても意図に反することが起きてから対応する形で記述する形になるため、ユーザの対応が増えてしまう危険性もある。本研究では、ユーザの要求を元にしたルールの作成を支援するため、自動的に提案することはできないが、ユーザが望まない状況に陥るということは無い。

6.3 定量評価

本節では定量評価として、ユーザが要求を出してから条件を出力するまでにかかった時間について計測し、データ量に応じて時間の増減を調べた。また、その際クラス属性として与える属性が離散値の場合、連続値の場合について評価した。

6.3.1 離散値を取るクラス属性

条件を作成する際に、決定木のクラス属性に該当する属性が離散値を取る場合、条件を出力するまでにかかる計算時間を評価した。データ数と計算時間の関係を図 6.4 に示す。

クラス属性が離散値を取った場合、ユーザの要求から条件を出力するまでに必要な計算量はデータ量 n に対して $O(n)$ と表現できた。この結果は、統計処理部と木作成部の計算量が $O(n)$ と表現でき、条件出力部の計算量が木の記述量に比例するため、木の複雑さが $\log n$ に比例するとした場合、 $O(\log n)$ と表現できることが理由として考えられる。統計処理部の処理内容としては、データの読み込みと形式変換が該当し、データ全体を最初から最後まで順に処理する内容となっている。また木作成部では、利用している C4.5 のアルゴリズムが、属性数を m と表現し、データ数を n と表現した場

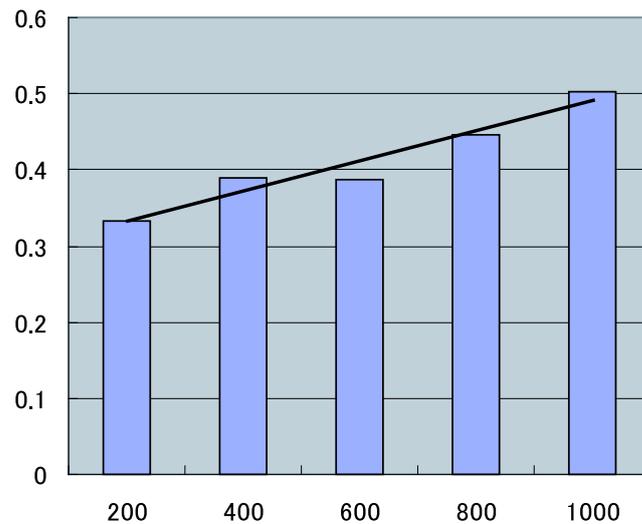


図 6.4: クラス属性が離散値を取る場合に条件出力にかかる時間

合, $O(mn)$ と表現できることが知られているため, データ量に対しては $O(n)$ と表現できる.

6.3.2 連続値を取るクラス属性

条件を作成する際に, 決定木のクラス属性に該当する属性が連続値を取る場合, 条件を出力するまでにかかる計算時間を評価した. データ数と計算時間の関係を図 6.4 に示す.

クラス属性が連続値を取った場合, ユーザの要求から条件を出力するまでに必要な計算量はデータ量 n に対して $O(n^2)$ と表現できた. この結果は, 離散化のために統計処理部でそれぞれのデータ間の距離を求め, 最小距離のデータを結合する処理を繰り返し行うことが理由として考えられる. この処理は, データ数を n として k 個に離散化を行う場合, 一回ごとにデータ数を減らしながら $n - k$ 回繰り返されるため, 理論上は $\sum_{i=0}^{n-k} i$ となり, 計算量は $O(n^2)$ となる.

6.4 本章のまとめ

本章では QuPen の動作実験と評価について述べた. 次章で本論文をまとめる.

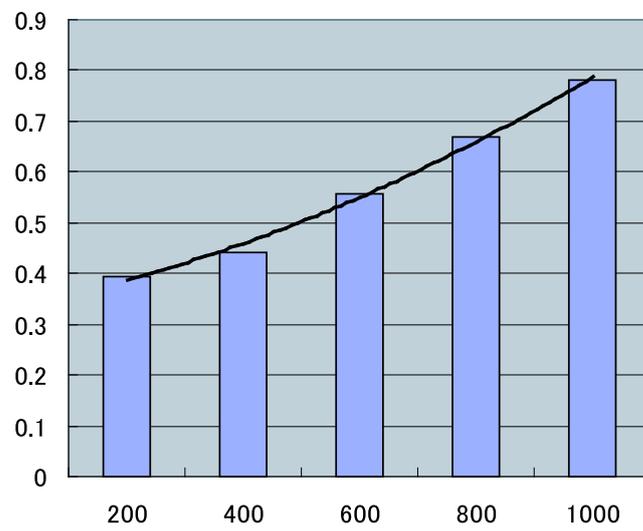


図 6.5: クラス属性が連続値を取る場合に条件出力にかかる時間

第7章 結論

本章では、今後の課題について述べ、最後に本論文をまとめる。

7.1 今後の課題

本システムにおける、今後の課題として、多様なセンサや機器を用いた評価、ルールに基づくコンテキストウェアサービス実現フレームワークとの連携、離散化の計算量改善、ルールの因果関係記述を挙げる。

7.1.1 多様なセンサや機器を用いた評価

今回の実験では、センサと機器をそれぞれ2種類用いた実験を行ったが、簡単な条件の提案に留まった。今後、ユビキタス環境を想定した実験施設を用いて、多様なセンサや機器を多数用いて、長時間のデータ収集を行い、日時を用いた周期的な行動パターンや、人の手では条件記述が難しいと認められる条件の記述を目指す。

7.1.2 ルールに基づくコンテキストウェアサービス実現フレームワークとの連携

本研究が対象とする、ルールに基づくコンテキストウェアサービス実現フレームワークに、本システムから提案を受けて作成したルールを記述し、実際に連携させた動作実験を行いたい。また、その結果を分析し、得られた条件に対して、自動記述に近づけられる条件、ユーザに確認を求める条件、ユーザに提示せずに切り捨てられる条件などの、条件に対する合目的性を評価したい。

7.1.3 離散化の計算量改善

離散化の際に、データ全体で繰り返し距離算出を行うことで、計算量が $O(n^2)$ となっていたが、結合したデータ以外の距離は変化しないため、距離の計算結果をキャッシュしておくことで、計算量が減少すると考えられる。また、現在1つずつ順番にデータを結合していくが、いくつか結合できそうなデータを同時に結合することで効率化ができると考えられる。

7.1.4 ルールの因果関係記述

本システムが提案する条件は、ユーザが指定した属性の変化と因果関係が強い属性を用いて表現されるが、どちらが原因となり、どちらが結果となるかが不明確である。例えば、人が部屋にいるから電気がついている、と表現される場合もあり、電気がついているから人が部屋にいる、と表現される場合もある。この因果関係の記述ができれば、条件の可読性も増すと考えられる。

7.2 まとめ

本論文では、ルールに従って動作するコンテキストウェアアプリケーションに対し、ユーザの要求からルールを提案する機構、QuPen について述べた。現在、コンテキストウェアサービスのルールを記述する方式の多くは、ユーザがサービスを動作させる状況について考慮する必要がある。これでは、センサや機器が今後増加した場合、ユーザがルールを記述する際に考慮すべきセンサや機器の情報は増加し、ルールの作成が困難になる。

本論文では、決定木学習にクラス属性を離散化する手法を追加し、ユーザが興味を抱いた状況がどのようなコンテキストで表現できるかを条件として記述するシステム、QuPen を構築した。QuPen はユーザの要求であるサービスの動作から、サービスを受けていた状況を条件として作成し、ユーザに提案する。これにより、ユーザは実現したいサービスを考慮するだけで、コンテキストウェアアプリケーションのルールを作成できる。

謝辞

本研究の機会を与えてくださり，御指導を賜りました慶応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，重要な御助言を頂きました，慶應義塾大学政策・メディア研究科助教授高汐一紀博士の御指導に深く感謝致します．

慶応義塾大学徳田・村井・楠本・中村・南合同研究会の先輩方には折りにふれ貴重な示唆や御助言，御指導を頂きました．特に，徳田研究室の先生方や先輩方，ACE研究グループ，move!研究グループの方々に深く感謝いたします．

また，慶應義塾大学政策・メディア研究科博士課程の門田昌哉氏には，丁寧な御指導をを賜りました．ここに深い感謝の念を表します．

また，大澤亮氏，駒木亮伯氏，榊原寛氏，宮島信吾氏，米沢拓郎氏，小泉健吾氏の協力と心遣いに感謝致します．

最後に，研究の日々を共に過ごした，幸田拓耶氏，丸山大佑氏その他多くの友人に深く感謝し，謝辞と致します．

平成 18 年 2 月 14 日
出内 将夫

参考文献

- [1] Intel exploratory research. <http://www.intel.com/research/>.
- [2] Massachusetts institute of technology. <http://web.mit.edu/>.
- [3] Service discovery in sensor networks: An overview.
- [4] Telecooperation office (teco).
<http://www.teco.edu/>.
- [5] 産業技術総合研究所. <http://www.aist.go.jp/>.
- [6] Daniel Salber Anind K. Dey and Gregory D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. In *Anchor article of a special issue on context-aware computing in the Human-Computer Interaction*, Vol. 16, pp. pp. 97–166, 2001.
- [7] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive*, pp. 1–17, 2004.
- [8] Chris Beckmann, Sunny Consolvo, and Anthony LaMarca. Some assembly required: Supporting end-user sensor installation in domestic ubiquitous computing environments. In *Ubicomp*, pp. 107–124, 2004.
- [9] M. Beigl and H. Gellersen. Smart-its: An embedded platform for smart objects, 2003.
- [10] Michael Beigl, Christian Decker, Albert Krohn, Till Riedel, and Tobias Zimmer. μ parts: Low cost sensor networks at scale. In *7th International Conference on Ubiquitous Computing Demonstration Proceedings (Ubicomp 05)*, 2005.
- [11] LTD. CHUO ELECTRONICS CO. ネットワーク対応電源制御装置 nd-pc01 .
<http://www.cec.co.jp/CEC/products/net/netdev/nd-pc01/index.html>.
- [12] Brian Clarkson. Life patterns: structure from wearable sensors. In *PhD thesis, Massachusetts Institute of Technology*, 2002.
- [13] C. Meek D. Chickering, D. Heckerman. A bayesian approach to learning bayesian networks with local structure. In *In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence, Providence*, pp. 80–89, 8 1997.

- [14] K. Rommelse D. Heckerman, J. Breese. Decision-theoretic troubleshooting. In *CACM*, pp. 49–57, 1995.
- [15] Anind K. Dey and Tim Sohn. Supporting end user programming of context-aware applications. In *CHI2003, end user development workshop*, 2004.
- [16] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, pp. 194–202, 1995.
- [17] Howard Hamilton, Ergun Gurak, Leah Findlater, and Wayne Olive. C4.5 tutorial.
<http://www2.cs.uregina.ca/~hamilton/courses/831/notes/ml/dtrees/c4.5/tutorial.html>.
- [18] S. Harabagiu, G. Miller, and D. Moldovan. Wordnet 2 - a morphologically and semantically enhanced resource, 1999.
- [19] Robert K. Harle and Andy Hopper. Using personnel movements for indoor autonomous environment discovery. In *PerCom*, pp. 125–132, 2003.
- [20] Chikio Hayashi. The quantitative study of national character: Interchronological and international perspectives. In *International Journal of Cultural Studies*, pp. 91–114, 1998.
- [21] J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization, 2001.
- [22] J.J. Hopfield and D.W. Tank. Neural computation of decisions optimization problems. In *Biological Cybernetics*, pp. 141–152, 1985.
- [23] Stephen S. Intille, Kent Larson, J. S. Beaudin, J. Nawyn, E. Munguia Tapia, and P. Kaushik. A living laboratory for the design and evaluation of ubiquitous computing technologies. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pp. 1941–1944, New York, NY, USA, 2005. ACM Press.
- [24] R. K. Jain. John Wiley & Sons, Inc., 1991.
- [25] Fahim Kawsar Kaori Fujinami and Tatu Nakajima. Awaremirror: A personalized display using a mirror. In *The 3rd International Conference on Pervasive Computing*, pp. pp. 315–332, 5 2005.
- [26] Cory D. Kidd, Robert Orr, Gregory D. Abowd, Christopher G. Atkeson, Irfan A. Essa, Blair MacIntyre, Elizabeth D. Mynatt, Thad Starner, and Wendy Newstetter. The aware home: A living laboratory for ubiquitous computing research. In *Cooperative Buildings*, pp. 191–198, 1999.

- [27] UAH Earth System Science Lab. Sensor modeling language.
<http://stromboli.nsstc.uah.edu/SensorML/>.
- [28] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*, pp. 237–245, London, UK, 2002. Springer-Verlag.
- [29] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. In *The Psychological Review, 1956, vol. 63*. 81-97, 1956.
- [30] Michael C. Mozer. The neural network house: An environment that adapts to its inhabitants. In *Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, pp. 110–114, 1998.
- [31] NICT. ユビキタスホーム実証実験.
http://www2.nict.go.jp/jt/a135/research/ubiquitous_home.html.
- [32] Yasuyoshi Inagaki Nobuo Kawaguchi, Toshikazu Miyazaki. ユビキタス情報環境における履歴を用いた機器操作支援手法. 情報処理学会ユビキタスコンピューティング研究会/映像情報メディア学会技術報告, 2004.
- [33] J. Ross Quinlan. Discovering rules from large collections of examples: a case study. In *Expert Systems in the Microelectronic Age*. Edinburgh University Press, 1979.
- [34] J. Ross Quinlan. Induction of decision trees. In *Machine Learning vol.1*, pp. 81–106. Kluwer Academic Press, 1986.
- [35] Dimitrios Raptis, Nikolaos Tselios, and Nikolaos Avouris. Context-based design of mobile applications for museums: a survey of existing practices. In *MobileHCI '05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pp. 153–160, New York, NY, USA, 2005. ACM Press.
- [36] T. Antti T. Urpo L.V. Kristof V.V. Walter S. Albrecht, K.A. Aidoo. Advanced interaction in context. In *1st International Symposium on Handheld and Ubiquitous Computing (HUC99)*, pp. pp. 89–101, 1999.
- [37] Ltd. SECOM Co. Secom town ホームセキュリティについて.
http://www.secomtown.com/hs/about/m_gaiyo.asp.
- [38] Claude E. Shannon. A mathematical theory of communication. In *The Bell System Technical Journal, 27*, pp. 379–423. Edinburgh University Press, 1948.

- [39] Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Andreas Krause, Neema Moraveji, Kathryn Reiger, Jeremy Shaffer, and Fei Lung Wong. Sensay: A context-aware mobile phone. In *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, p. 248, Washington, DC, USA, 2003. IEEE Computer Society.
- [40] InData Systems. Rf code spider rfid system. <http://www.indatasys.com/html/products>.
- [41] Shuhei Tamaru, Jin Nakazawa, Kazunori Takashio, and Hideyuki Tokuda. Ppnp: A privacy profile negotiation protocol for services in public spaces. In *Proceedings of Fifth International Conference on Ubiquitous Computing (UbiComp2003) First International Workshop on Ubiquitous Systems for Supporting Social Interaction and Face-to-Face Communication in Public Spaces*, 10 2003.
- [42] T.Okoshi, S.wakayama, Y.Sugita, S.Aoki, T.Iwamoto, J.Nakazawa, D.Furusaka, M.Iwai, A.Kusumoto, N.Harashima, J.Yura, N.Nishio, Y.Tobe, Y.Ikeda, and H.Tokuda. Smart space laboratory project: Toward the next generation computing environment. In *Proceedings of IEEE International Workshop on Networked Appliances(IWNA)*, pp. ??–, 2 2001.
- [43] M Weiser. The computer for the 21st century. In *Scientific American 256(3),94–104, September 1991*.
- [44] R. J. Wilson. グラフ理論. 近代科学社, 1992.
- [45] [RFC2245] Network working group. Internet calendaring and scheduling core object specification (icalendar), 1998.
- [46] Tao Zhang. An architecture for building customizable context-aware applications by end-users. In *The 2nd International Conference on Pervasive Computing, doctoral colloquium*, pp. ??–??, 5 2004.
- [47] ユビキタスネットワークワーキングフォーラム. ユビキタスネットワークの実現に向けて. Oct 2004.
- [48] 猿渡 俊介 川原 圭博 南 正輝 森川 博之 青山 友紀 篠田 庄司 永原 崇範. ユビキタス環境に向けたセンサネットワークアプリケーション構築支援のための開発用モジュール u³ (u-cube) の設計と実装. 電子情報通信学会技術研究報告, 第 Vol.102No.392 巻, pp. 61–66, 7 2003.
- [49] 川原圭博, 司化, 森川博之, 青山友紀. コンテキストウェアサービス実現に関する一検討. 電子情報通信学会ソサイエティ大会, B-20-7, September 2004.

- [50] 古川康一, 尾崎知伸, 植野研. 帰納論理プログラミング. 共立出版, 2001.
- [51] 高林哲, 小松弘幸, 増井俊之. Migemo: 日本語のインクリメンタル検索, 2002.
- [52] 米澤拓郎, 小泉健吾, 守分滋, 永田智大, 徳田英幸. Smart furniture 間の柔軟なサービスローミングを実現するミドルウェアの構築. 情報処理学会第三回ユビキタスコンピューティングシステム研究会論文集, 第 2004 巻, pp. 39–46, 1 2004.
- [53] 本村陽一. 乳幼児の事故予防のための知識データベースと事故予測モデル.
<http://staff.aist.go.jp/y.motomura/dh/2004akachan.files/frame.html>.
- [54] 圓川隆夫. 多変量のデータ解析. 朝倉書店, 1988.