

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2019-169147
(P2019-169147A)

(43) 公開日 令和1年10月3日(2019.10.3)

(51) Int. Cl.

G06N 20/00 (2019.01)

F I

G06N 20/00

テーマコード(参考)

審査請求 未請求 請求項の数 10 O L (全 31 頁)

<p>(21) 出願番号 特願2019-50616 (P2019-50616)</p> <p>(22) 出願日 平成31年3月19日 (2019. 3. 19)</p> <p>(31) 優先権主張番号 特願2018-53454 (P2018-53454)</p> <p>(32) 優先日 平成30年3月20日 (2018. 3. 20)</p> <p>(33) 優先権主張国・地域又は機関 日本国 (JP)</p>	<p>(71) 出願人 504133110 国立大学法人電気通信大学 東京都調布市調布ヶ丘一丁目5番地1</p> <p>(74) 代理人 110000925 特許業務法人信友国際特許事務所</p> <p>(72) 発明者 沼尾 雅之 東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内</p> <p>(72) 発明者 大石 伸之 東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内</p> <p>(72) 発明者 永間 慎太郎 東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内</p>
---	---

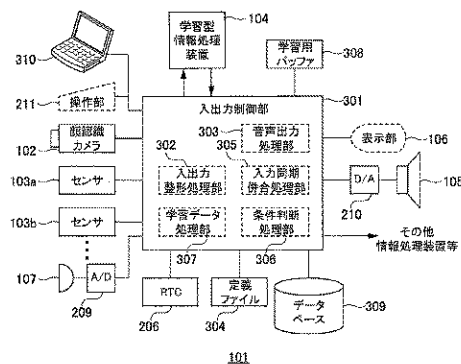
(54) 【発明の名称】 情報処理装置及び情報処理システム

(57) 【要約】

【課題】システム構築と変更が容易であり、多種多様なセンサに柔軟に対応でき、また少量のデータで効率的に学習が実現でき、推定精度も高い情報処理装置を提供する。

【解決手段】情報処理装置は、種々のセンサから得られる各々のイベントの内容を、入力同期併合処理部、学習データ処理部を通じて、条件判断処理部に引き渡す。これらは定義ファイルにて複数回使用することが可能であり、これらを組み合わせることで、複雑なデータフローを容易に形成し、多種多様なA/D Lの識別あるいは推定を行うことができる。

【選択図】 図3



【特許請求の範囲】**【請求項 1】**

時系列データを所定の時間間隔毎に区切り、日時情報とタグ名を付したイベントとして出力する入出力整形処理部と、

主入力側のイベントと副入力側のイベントを併合し、日時情報とタグ名を付した単一のイベントに変換して出力する入力同期併合処理部と、

単一のイベントに含まれるデータに対し、所定の条件判断を行い、条件判断の結果に応じて異なるイベントを出力する条件判断処理部と、

単一のイベントに含まれるデータを特徴ベクトルとして学習型情報処理装置に転送し、受信した推定結果を単一のイベントとして出力する学習データ処理部と、

複数のセンサが出力するデータを前記入出力整形処理部によって複数種類のイベントに変換した後、前記入力同期併合処理部と、前記条件判断処理部と、前記学習データ処理部を組み合わせ、前記複数のセンサが計測及び/または観測した事象の識別または推定を実施する入出力制御部と

を具備する情報処理装置。

【請求項 2】

前記学習型情報処理装置は、前記学習データ処理部より前記特徴ベクトルを受信し、学習アルゴリズムに基づく推定処理を行い、前記推定結果を前記学習データ処理部に送信するものであり、

オラクルに問い合わせを行うことでイベントに対する正解ラベルを取得する概念学習器を更に備え、

前記情報処理装置の前記学習データ処理部は、前記推定結果に基づいて観察対象の状態または動作を特定できなかった場合には、特定できなかった旨のタグを付したイベントを前記概念学習器へ送信することで、前記概念学習器より正解ラベルを取得し、前記正解ラベルが認識済みの状態または動作である場合には、前記学習型情報処理装置に対して既知の概念に対する再学習をさせることで、定義ファイルの既知概念を拡張する更新を行い、前記正解ラベルが未だ認識していない状態または動作である場合には、未知の概念を獲得するための新規の学習をさせることで、定義ファイルに新規の概念を追加する更新を行うことを特徴とする、

請求項 1 に記載の情報処理装置。

【請求項 3】

前記入出力制御部は、定義ファイルに記述されたディレクティブに従い、イベントの処理に際し、前記入力同期併合処理部と、前記条件判断処理部と、前記学習データ処理部をそれぞれ複数個配置することが可能であることを特徴とする、

請求項 1 または 2 に記載の情報処理装置。

【請求項 4】

前記入出力制御部は、複数のイベントを処理するに際して、イベントの属するシナリオグループの優先度に応じて前記イベントの処理を優先して行うことを特徴とする、

請求項 3 に記載の情報処理装置。

【請求項 5】

前記入出力整形処理部は、JSONフォーマットのイベントに、日時情報とタグ名を付すことを特徴とする、

請求項 3 または 4 に記載の情報処理装置。

【請求項 6】

情報処理装置と学習型情報処理装置と備える情報処理システムであって、

前記情報処理装置は、

時系列データを所定の時間間隔毎に区切り、日時情報とタグ名を付したイベントとして出力する入出力整形処理部と、

主入力側のイベントと副入力側のイベントを併合し、日時情報とタグ名を付した単一のイベントに変換して出力する入力同期併合処理部と、

単一のイベントに含まれるデータに対し、所定の条件判断を行い、条件判断の結果に応じて異なるイベントを出力する条件判断処理部と、

単一のイベントに含まれるデータを特徴ベクトルとして前記学習型情報処理装置に転送し、受信した推定結果を単一のイベントとして出力する学習データ処理部と、

複数のセンサが出力するデータを前記入出力整形処理部によって複数種類のイベントに変換した後、前記入力同期併合処理部と、前記条件判断処理部と、前記学習データ処理部を組み合わせ、前記複数のセンサが計測及び/または観測した事象の識別または推定を実施する入出力制御部と

を具備し、

前記学習型情報処理装置は、前記学習データ処理部より前記特徴ベクトルを受信し、学習アルゴリズムに基づく推定処理を行い、前記推定結果を前記学習データ処理部に送信することを特徴とする情報処理システム。

10

【請求項 7】

オラクルに問い合わせを行うことでイベントに対する正解ラベルを取得する概念学習器を更に備え、

前記情報処理装置の前記学習データ処理部は、前記推定結果に基づいて観察対象の状態または動作を特定できなかった場合には、特定できなかった旨のタグを付したイベントを前記概念学習器へ送信することで、前記概念学習器より正解ラベルを取得し、前記正解ラベルが認識済みの状態または動作である場合には、前記学習型情報処理装置に対して既知の概念に対する再学習をさせることで、定義ファイルの既知概念を拡張する更新を行い、前記正解ラベルが未だ認識していない状態または動作である場合には、未知の概念を獲得するための新規の学習をさせることで、定義ファイルに新規の概念を追加する更新を行うことを特徴とする、

20

請求項 6 に記載の情報処理システム。

【請求項 8】

前記情報処理装置の前記入出力制御部は、定義ファイルに記述されたディレクティブに従い、イベントの処理に際し、前記入力同期併合処理部と、前記条件判断処理部と、前記学習データ処理部をそれぞれ複数個配置することが可能であることを特徴とする、

請求項 6 または 7 に記載の情報処理システム。

【請求項 9】

前記情報処理装置の前記入出力制御部は、複数のイベントを処理するに際して、イベントの属するシナリオグループの優先度に応じて前記イベントの処理を優先して行うことを特徴とする、

30

請求項 8 に記載の情報処理システム。

【請求項 10】

前記情報処理装置の前記入出力整形処理部は、JSONフォーマットのイベントに、日時情報とタグ名を付すことを特徴とする、請求項 8 または 9 に記載の情報処理システム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、多種多様な情報を受け入れて所定の出力を得る情報処理装置及び情報処理システムに関する。

40

【背景技術】

【0002】

高齢者の人口は世界中で急増している。医学の進歩などにより人の寿命が伸びる一方、全人口に占める労働人口の割合は減少し、介護・医療スタッフの不足を始めとした様々な問題が生じている。これらの問題を解決するために、AAL (Ambient Assisted Living : 自立生活支援技術) が大きな関心を集めている。

【0003】

AAL は、我々を取り巻く AI (Artificial Intelligence : 人工知能) ・ IoT (Int

50

ernet of Things :モノのインターネット) 技術を駆使し、高齢者の自立した生活を可能な限り持続させることで、QOL (Quality of Life :生活の質) の向上をサポートしていくことである。AALデバイスやサービスを適切に使用することで、廃用症候群等の予防や、リハビリの補助、また日常生活のサポートを通して、高齢者のQOLを維持する、もしくは改善する、といったことが期待される。

【0004】

AAL研究者において特に大きな注目を集めている話題の一つが、ADL (Activities of Daily Living :日常生活行動) の観察である。AI・IoT技術を活用し、高齢者のADLをモニタリングし、結果に応じたフィードバックを投げかけることで、廃用症候群の防止、またリハビリ進展度の判断等に役立つことが期待される。

10

【0005】

特許文献1には、顔の表情と動作を認識可能な顔表情動作認識方法及び装置が開示されている。

特許文献2には、効率良く多職種間の連携を支援する、連携支援装置、連携支援システム、連携支援方法、及びプログラムが開示されている。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特開2017-162409号公報

【特許文献2】特開2016-212925号公報

20

【発明の概要】

【発明が解決しようとする課題】

【0007】

ADLの観察には、AI等の学習アルゴリズムを用いる。学習アルゴリズムを実行する計算機に、観察対象の状態や行動等を計測した情報を入力し、観察対象の状態や行動を推定する。

これまで、学習アルゴリズムを用いた事象認識技術は、入力されるデータの形態に柔軟性がなく固定的なものであった。すなわち、計算機で実行されるプログラムは固定的に構築されているため、装置が完成した後から、例えば新たなセンサ等の新たな種類の情報を追加することが困難であった。

30

【0008】

また、現在主流の学習アルゴリズムを用いた情報処理装置は、バッチ学習と呼ばれる、一度に大量のデータを情報処理装置に与えて学習させる手法が採用されている。しかし、多くの事象において、統一されたフォーマットのデータを大量に入手することは困難である。ADLの観察においても同様で、ある観察対象者から大量のデータを得ることは困難である場合が多い。

【0009】

本発明はかかる課題に鑑みてなされたものであり、システムの構築と変更を容易に行うことにより多種多様なセンサに柔軟に対応可能とし、また少量のデータで効率的な学習を実現可能な、推定精度も高い情報処理装置及び情報処理システムを提供することを目的とする。

40

【課題を解決するための手段】

【0010】

上記課題を解決するために、本発明の情報処理装置は、時系列データを所定の時間間隔毎に区切り、日時情報とタグ名を付したイベントとして出力する入出力整形処理部と、主入力側のイベントと副入力側のイベントを併合し、日時情報とタグ名を付した単一のイベントに変換して出力する入力同期併合処理部と、単一のイベントに含まれるデータに対し、所定の条件判断を行い、条件判断の結果に応じて異なるイベントを出力する条件判断処理部とを具備する。

【0011】

50

更に、本発明の情報処理装置は、単一のイベントに含まれるデータの特徴ベクトルとして学習型情報処理装置に転送し、受信した推定結果を単一のイベントとして出力する学習データ処理部と、複数のセンサが出力するデータを入出力整形処理部によって複数種類のイベントに変換した後、入力同期併合処理部と、条件判断処理部と、学習データ処理部を組み合わせ、複数のセンサが計測及び/または観測した事象の識別または推定を実施する入出力制御部とを具備する。

を具備する。

【発明の効果】

【0012】

本発明によれば、システムの構築と変更を容易に行うことができるので、多種多様なセンサに柔軟に対応することができる。また、少量のデータで効率的な学習を実現することができ、推定精度も高い情報処理装置及び情報処理システムを提供することができる。

上記した以外の課題、構成及び効果は、以下の実施形態の説明により明らかにされる。

【図面の簡単な説明】

【0013】

【図1】本発明の実施形態に係る情報処理装置の全体構成を示すブロック図である。

【図2】情報処理装置のハードウェア構成を示すブロック図である。

【図3】情報処理装置のソフトウェア機能を示すブロック図である。

【図4】入力同期併合処理部と、条件判断処理部と、学習データ処理部の入出力を説明するブロック図である。

【図5】情報処理装置が実行するデータ処理のアーキテクチャを示す概念図である。

【図6】マルチモーダルセンサからイベントを取得して、センサレベルデータフローでイベントの加工を行うことで観察対象者の行動認識を行い、行動レベルデータフローにイベントを出力する過程を示す図である。

【図7】マルチモーダルセンサからイベントを取得して、センサレベルデータフローでイベントの加工を、概念レベルデータフローにてイベントの推定を行うことで観察対象者の行動認識を行い、行動レベルデータフローにイベントを出力する過程を示す図である。

【図8】マルチモーダルセンサからイベントを取得して、センサレベルデータフローでイベントの加工を、概念レベルデータフローにてイベントの推定を行うことで観察対象者の行動認識を行い、概念学習器と対話して学習処理を行う過程を示す図である。

【図9】マルチモーダルセンサからイベントを取得して、センサレベルデータフローにおける認識結果と、概念レベルデータフローにおける推定結果とを組み合わせ、複数のADLを識別し、多彩な応答を行う過程を示す図である。

【図10】入出力制御部及び学習型情報処理装置の、データ分類モードにおける機能ブロック図である。

【図11】入出力制御部及び学習型情報処理装置の、新規クラス学習モードにおける機能ブロック図である。

【図12】入出力制御部及び学習型情報処理装置の新規クラス学習モードにおいて、概念学習器が情報処理装置とオラクルと学習型情報処理装置に対して実行するデータ処理の流れを示す概略図である。

【図13】情報処理装置における、定義ファイルの更新処理を示すブロック図である。

【図14】本発明の変形例に係る情報処理装置のソフトウェア機能を示すブロック図である。

【図15】シリアライザの機能ブロック図である。

【図16】シリアライザの動作の流れを示すフローチャートである。

【図17】異常対応シナリオ処理の動作の流れを示すフローチャートである。

【図18】健康チェックシナリオ処理、リラクゼーションシナリオ処理、応答シナリオ処理の動作の流れを示すフローチャートである。

【図19】キューの状態を説明するための概略図である。

【発明を実施するための形態】

10

20

30

40

50

【 0 0 1 4 】

[情報処理装置 1 0 1 : 全体構成]

図 1 は、本発明の実施形態に係る情報処理装置 1 0 1 の全体構成を示すブロック図である。

情報処理装置 1 0 1 は、例えば、観察対象者である独居高齢者や老人施設における高齢者が適切に生活を営んでいるのかを観察する。また、情報処理装置 1 0 1 は独居高齢者に対して擬似的な会話を行うことで、独居高齢者が廃用症候群等に罹患することを防止し、以て独居高齢者の Q o L を改善する。

【 0 0 1 5 】

情報処理装置 1 0 1 には、顔認識カメラ 1 0 2、赤外線センサや非接触温度センサ等の種々のセンサ 1 0 3 a、1 0 3 b が接続されている。情報処理装置 1 0 1 はこれらセンサから、観察対象者を計測した結果得られる情報（時系列データ）を受信する。そして、観察対象者の照合結果に応じて、観察対象者の A D L を判断、あるいは推定する。

【 0 0 1 6 】

情報処理装置 1 0 1 は推定の際、学習アルゴリズムに基づく推定処理を、ネットワークを通じて外部の学習型情報処理装置 1 0 4 に依頼して、推定結果を受信する。そして情報処理装置 1 0 1 は推定結果に従い、スピーカ 1 0 5 から所定の音声を発する。また、必要に応じて液晶ディスプレイ等の表示部 1 0 6 に、所定の情報を表示する、等の処理を行う。

【 0 0 1 7 】

[情報処理装置 1 0 1 : ハードウェア構成]

図 2 は、情報処理装置 1 0 1 のハードウェア構成を示すブロック図である。

情報処理装置 1 0 1 は、バス 2 0 1 に接続された、CPU 2 0 2、ROM 2 0 3、RAM 2 0 4、不揮発性ストレージ 2 0 5 を備える。

バス 2 0 1 には更に、現在日時情報を出力するリアルタイムクロック（図 2 中「R T C」と表記、以下「R T C」と略）2 0 6、ネットワークインターフェース（図 2 中「N I C」と表記、以下「N I C」と略）2 0 7、シリアルインターフェース（図 2 中「シリアル I / F」と表記、以下「シリアル I / F」と略）2 0 8 を通じて顔認識カメラ 1 0 2、センサ 1 0 3 a、1 0 3 b 等が、A / D 変換器 2 0 9 を通じてマイク 1 0 7 が、D / A 変換器 2 1 0 を通じてスピーカ 1 0 5 が、それぞれ接続されている。

また必要に応じて、表示部 1 0 6 及び操作部 2 1 1 がバス 2 0 1 に接続されることがある。

【 0 0 1 8 】

情報処理装置 1 0 1 は、英国ラズベリーパイ財団（<http://www.raspberrypi.org/>）が開発する「R a s p b e r r y P i」等を始めとする、近年普及している安価なシングルボードコンピュータが利用可能である。

学習型情報処理装置 1 0 4 は、十分な演算能力を有するパソコン等が利用可能である。但し、学習型情報処理装置 1 0 4 において実行される学習アルゴリズムが要求する演算量が少ない場合は、情報処理装置 1 0 1 に内蔵させてもよい。

【 0 0 1 9 】

[情報処理装置 1 0 1 : ソフトウェア機能]

図 3 は、情報処理装置 1 0 1 のソフトウェア機能を示すブロック図である。

入出力制御部 3 0 1 には、顔認識カメラ 1 0 2、赤外線センサや非接触温度センサ等の種々のセンサ 1 0 3 a、1 0 3 b が接続されている。また、A / D 変換器 2 0 9 を通じて観察対象者の音声を認識するためのマイク 1 0 7 と、D / A 変換器 2 1 0 を通じて観察対象者に会話音声を発するためのスピーカ 1 0 5 も接続されている。

顔認識カメラ 1 0 2 や種々のセンサは、情報処理装置 1 0 1 におけるマルチモーダルセンサ（multimodal sensor）として扱われる。すなわち、観察対象者や観察対象とする事象を、映像、音声、温度、加速度等、様々な物理量等で計測するセンサ群としてみなされる。

10

20

30

40

50

【 0 0 2 0 】

情報処理装置 1 0 1 の主たる機能は、入出力制御部 3 0 1 に集約されている。この入出力制御部 3 0 1 は、サンプリングクロック毎、あるいは所定の時間間隔毎にセンサから得られる情報に、オブジェクト指向の思想に基づき、種々の情報を付加する。種々の情報が付加された情報をイベントと呼ぶ。入出力制御部 3 0 1 の大部分の機能は、オープンソースのデータ収集ソフトウェアである Fluentd (<https://www.fluentd.org/>) に、後述する情報処理機能をプラグインとして追加することで実現している。

また、入出力制御部 3 0 1 には、後述する概念学習器として機能する、パソコン等の外部情報処理装置 3 1 0 が、ネットワークを通じて接続される。

【 0 0 2 1 】

センサから得られる情報をイベントに格納する際のルールは、センサの種類によって異なる。

例えば温度や湿度、加速度等の一般的な物理現象を測定するセンサの場合は、A / D 変換器 2 0 9 のサンプリングクロック毎にデータを取得すべきである。

一方、例えばマイク 1 0 7 等から得られる音声データのように、ある程度連続した時間間隔のデータが必要になる場合は、A / D 変換器 2 0 9 のサンプリングクロック毎にデータを取得しても、どのような音が鳴動しているのかわからない。したがってマイク 1 0 7 の場合には、無音検出等の周知の技術を用いて、会話等の推定が可能な程度の長さの音声データを取得することが好ましい。

【 0 0 2 2 】

入出力制御部 3 0 1 の機能として備わっている入出力整形処理部 3 0 2 は、各種センサから得られる情報を、統一されたデータフォーマットに従うプレインテキストで記述されたイベントに変換する。また必要に応じて、イベントから内容を取り出して、表示部 1 0 6 に表示する内容や、音声出力処理部 3 0 3 を介して音声出力データに変換する。この統一されたデータフォーマットは、基本的には一定のルールさえ満たされていれば何でもよいのであるが、データの可搬性という観点から、JSON (JavaScript Object Notation (「 J A V A S C R I P T 」 は登録商標)) が望ましい。

【 0 0 2 3 】

以下に、JSON フォーマットにて記述されたイベントの一例を示す。

【 0 0 2 4 】

```
2017 11 03 19:35:32.000000000 +0900 sensor.microwave:
{"heart":79, "breath":12}
```

【 0 0 2 5 】

イベントは、「time tag:record」という記述順で記載される。「time」は POSIX time フォーマットの絶対日時情報、「record」は JSON 形式であるので、始め波括弧 ("{") で始まり、終わり波括弧 ("} ") で終わる。

「tag」は JSON 形式オブジェクトに与えられる名前 (以下「タグ名」) であり、オブジェクト指向の慣例に倣い、「name1.name2.name3 」というように、ピリオドで連結する。すなわち、ピリオドはタグ名のフィールドセパレータである。「time」と「tag」の間はスペース (" ") で区切られ、「tag」と「record」の間はコロン (":") で区切られる。

入出力整形処理部 3 0 2 はイベントを生成する際、RTC 2 0 6 からイベントに付すための絶対日時情報を読み取る。

【 0 0 2 6 】

入出力制御部 3 0 1 の機能として備わっている音声出力処理部 3 0 3 は、D / A 変換器 2 1 0 を通じてスピーカ 1 0 5 から観察対象者に対して会話音声を出力するためのデジタルオーディオデータを生成する、周知のスピーチプロセッサである。

【 0 0 2 7 】

入出力制御部 3 0 1 は、マルチモーダルセンサから種々のデータの入力を受けると、入出力整形処理部 3 0 2 によって前述の JSON 形式のイベントに変換した後、定義ファイ

10

20

30

40

50

ル 3 0 4 に記述されている内容に従い、入力同期併合処理部 3 0 5、条件判断処理部 3 0 6、学習データ処理部 3 0 7 にてイベントの加工等を行う。

定義ファイル 3 0 4 は例えば単一または複数のブレインテキストファイルである。

【 0 0 2 8 】

[入力同期併合処理部 3 0 5]

ここで図 4 A を参照して、入力同期併合処理部 3 0 5 について説明する。

図 4 A は、入力同期併合処理部 3 0 5 の入出力を説明するためのブロック図である。

入出力制御部 3 0 1 の機能として備わっている入力同期併合処理部 3 0 5 は、複数のイベントを受け入れて、各々のイベントの内容を単一のイベントに併合して出力する。単一のイベントを出力する際、入力同期併合処理部 3 0 5 は、定義ファイル 3 0 4 に記述することで指定されたタイミングで、イベントを出力する。イベント同士を併合するタイミングは、「before」、「after」、「simple」の 3 種類である。

10

【 0 0 2 9 】

定義ファイル 3 0 4 において併合タイミングに「before」が指定された場合には、入力同期併合処理部 3 0 5 は、定義ファイル 3 0 4 の「main tag」（主入力側）に記述された第一のイベントを受け取ると、第一のイベントを受け取った直前の、「sub tag」（副入力側）に記述された第二のイベントを、第一のイベントと共に併合して出力する。したがって、「before」の場合は、第一のイベントを受け取った日時が、出力されるイベントに付加される。

20

【 0 0 3 0 】

定義ファイル 3 0 4 において併合タイミングに「after」が指定された場合には、入力同期併合処理部 3 0 5 は、定義ファイル 3 0 4 の「main tag」に記述された第一のイベントを受け取ると、第一のイベントを受け取った直後の、「sub tag」に記述された第二のイベントを、第一のイベントと共に併合して出力する。したがって、「after」の場合は、第二のイベントを受け取った日時が、出力されるイベントに付加される。

【 0 0 3 1 】

定義ファイル 3 0 4 において併合タイミングに「simple」が指定された場合には、入力同期併合処理部 3 0 5 は、定義ファイル 3 0 4 の「main tag」に記述された第一のイベントを受け取ると、第一のイベントを受け取った直前の、「sub tag」に記述された第二のイベントを、第一のイベントと共に併合して出力する。また、第一のイベントを受け取った直後の、「sub tag」に記述された第二のイベントも、第一のイベントと共に併合して出力する。すなわち、「simple」は「before」と「after」の論理和である。したがって、「simple」の場合は、第一のイベントを受け取った時のみならず、第二のイベントを受け取った時にも、イベントを出力する。

30

【 0 0 3 2 】

以下に、定義ファイル 3 0 4 にて記述される入力同期併合処理部 3 0 5 の指示（ディレクティブ（directive））の一例を示す。定義ファイル 3 0 4 のディレクティブは、XML（eXtensible Markup Language）に類似するマークアップ言語の形式で記述される。

【 0 0 3 3 】

```
<match sensor.{temp, humid}.to merge>
```

40

```
  @type record merger
```

```
  tag sensor.temp humid
```

```
  main tag sensor.temp.to merge
```

```
  sub tag1 sensor.humid.to merge
```

```
  merge timing simple
```

```
<record>
```

```
  temp ${main["temp"]}
```

```
  humid ${sub1["humid"]}
```

```
</record>
```

```
</match>
```

50

【 0 0 3 4 】

「<match sensor.{temp, humid}.to merge>」は、ディレクティブの開始の宣言であると共に、「sensor.temp.to merge」タグが付されたイベントと「sensor.humid.to merge」タグが付されたイベントを、タグ名でマッチングさせることを示す。

「@type record merger」は、本ディレクティブが入力同期併合処理部 3 0 5 の処理を実行する「record merger」であることを示す。

「tag sensor.temp humid」は、本ディレクティブが出力するイベントのタグ名が「sensor.temp humid」であることを示す。

【 0 0 3 5 】

「main tag sensor.temp.to merge」は、前述の「main tag」である。すなわち、「sensor.temp.to merge」が第一のイベントである。 10

「sub tag1 sensor.humid.to merge」は、前述の「sub tag」である。すなわち、「sensor.humid.to merge」が第二のイベントである。

「merge timing simple」は、前述の併合タイミングである。

「<record>」から「</record>」で囲まれる中身は、出力するイベントの中に記述される要素を示す。これは JSON フォーマットで、「名前文字列:値」に変換される。

「</match>」は、「<match sensor.{temp, humid}.to merge>」ディレクティブの終了を示す。

【 0 0 3 6 】

[条件判断処理部 3 0 6] 20

次に図 4 B を参照して、条件判断処理部 3 0 6 について説明する。

図 4 B は、条件判断処理部 3 0 6 の入出力を説明するためのブロック図である。

入出力制御部 3 0 1 の機能として備わっている条件判断処理部 3 0 6 は、単一のイベントを受け入れて、イベントに含まれているデータに対し、所定の条件判断を行う。条件判断は、タグ名の一致または不一致、また数値データの、記述された閾値との比較結果である。条件判断は定義ファイル 3 0 4 内にて、プログラミング言語において一般的な「if..then...else...」の記述が可能である。

以下に、定義ファイル 3 0 4 にて記述される条件判断処理部 3 0 6 のディレクティブの一例を示す。

【 0 0 3 7 】 30

```
<match sensor.temp>
  @type condition checker
  tag1 sensor.room temp.is too cold
  tag2 sensor.room temp.is too hot
  tag else sensor.room temp.is good
  renew record false
  condition1 record ["temp"] <= 18 && record ["season"] == "winter"
  condition2 record ["temp"] >= 26 && record ["season"] == "winter"
  <record1>
    status "cold"
  </record1>
  <record2>
    status "hot"
  </record2>
  <record else>
    status "good"
  </record else>
</match>
```

40

【 0 0 3 8 】

「<match sensor.temp>」は、ディレクティブの開始の宣言であると共に、「sensor.te 50

mp」タグが付されたイベントを、タグ名でマッチングさせることを示す。

「@type condition checker」は、本ディレクティブが条件判断処理部 3 0 6 の処理を実行する「condition checker」であることを示す。

「tag1 sensor.room temp.is too cold」は、後述する「condition1」の条件に合致した場合において出力するイベントのタグ名が「sensor.room temp.is too cold」であることを示す。

【 0 0 3 9 】

「tag2 sensor.room temp.is too hot」は、後述する「condition2」の条件に合致した場合において出力するイベントのタグ名が「sensor.room temp.is too hot」であることを示す。

10

「tag else sensor.room temp.is good」は、condition1及びcondition2の何れにも当てはまらなかった場合において出力するイベントのタグ名が「sensor.room temp.is good」であることを示す。

「renew record false」は、受信したイベントに含まれるレコード（値）の内容を更新しないことを示す。

【 0 0 4 0 】

「condition1 record ["temp"] <= 18 && record ["season"] == "winter"」は、「condition1」の条件判断文である。「temp」の値が18以下（"<="）、且つ（"&&"）、「season」の値が「winter」であるか否かを判断する。

「condition2 record ["temp"] >= 26 && record ["season"] == "winter"」は、「condition2」の条件判断文である。「temp」の値が26以上（">="）、且つ（"&&"）、「season」の値が「winter」であるか否かを判断する。

20

【 0 0 4 1 】

「<record1>」から「</record1>」に囲まれる中身は、「condition1」の条件に合致した際に、イベントに「status:"cold"」というレコードを追加することを定義する。

同様に、「<record2>」から「</record2>」に囲まれる中身は、「condition2」の条件に合致した際に、イベントに「status:"hot"」というレコードを追加することを定義する。

そして、「<record else>」から「</record else>」に囲まれる中身は、「condition1」及び「condition2」の、何れの条件にも合致しなかった際に、イベントに「status:"good"」というレコードを追加することを定義する。

30

「</match>」は、「<match sensor.temp>」ディレクティブの終了宣言である。

【 0 0 4 2 】

[学習データ処理部 3 0 7]

次に図 4 C を参照して、学習データ処理部 3 0 7 について説明する。

図 4 C は、学習データ処理部 3 0 7 の入出力を説明するためのブロック図である。

学習データ処理部 3 0 7 は、入力されたイベントの内容を学習型情報処理装置 1 0 4 に特徴ベクトル（説明変数）として出力し、学習型情報処理装置 1 0 4 から目的変数を受け取り、イベントの形式に変換して出力する。

【 0 0 4 3 】

40

学習データ処理部 3 0 7 には、「train」と「analyze」という2種類のモードが存在する。

trainモードでは入力されるイベントを、ラベルと Datum というキーバリュ形式のデータの組に変換し、学習型情報処理装置 1 0 4 に送信することで、学習型情報処理装置 1 0 4 における学習モデルの訓練（学習）を行う。

本発明の実施形態に係る情報処理装置 1 0 1 は、学習型情報処理装置 1 0 4 の一例として、オンライン機械学習向け分散処理フレームワークである Jubatus (<http://jubat.us/ja/>) を使用している。Datum とは、この Jubatus が解釈可能な key-value データ形式を指す。

【 0 0 4 4 】

50

analyzeモードでは入力されるイベントをDatum形式に変換して学習型情報処理装置104に送信することで、分類・推薦・異常検知等、学習型情報処理装置104が提供している機能を使用して分析した結果をリアルタイムで受け取ることができる。

以下に、定義ファイル304にて記述される、学習データ処理部307のanalyzeモードにおけるディレクティブの一例を示す。

【0045】

```
<match concept.accelerometer.analyze>
  @type active online learner
  host <jubatus server's ip address>
  client api 'classifier'
  train analyze analyze
  num keys ax mean,ay mean,az mean,ax var,ay var,az var
  threshold 9.0
  tag for train concept.unclassifiable
</match>
```

10

【0046】

「<match concept.accelerometer.analyze>」は、ディレクティブの開始の宣言であると共に、「concept.accelerometer.analyze」タグが付されたイベントを、タグ名でマッチングさせることを示す。

「@type active online learner」は、本ディレクティブが学習データ処理部307の処理を実行する「active online learner」であることを示す。

20

「host <jubatus server's ip address>」は、学習型情報処理装置104のIPアドレスまたはホスト名（FQDN:Fully Qualified Domain Name、完全修飾ドメイン名）が記述される。

「client api 'classifier'」は、学習型情報処理装置104のクライアント側アプリケーションプログラムインターフェース、すなわち学習データ処理部307に対し、「分類器」として動作すべく指示することを示す。

【0047】

「train analyze analyze」は、学習データ処理部307に対し、analyzeモードで動作すべく指示することを示す。

30

「num keys ax mean,ay mean,az mean,ax var,ay var,az var」は、学習型情報処理装置104に引き渡すデータ、すなわち特徴ベクトルの一例を示す。この例では、3軸センサの各軸方向の平均値と、3軸センサの各軸方向にて検出した値が列挙されている。

「threshold 9.0」は、学習型情報処理装置104から受信した値を判定する際の閾値を示す。

「tag for train concept.unclassifiable」は、学習型情報処理装置104から受信した値が全て閾値に満たなかった場合に、出力するイベントに付すタグ名である。

「</match>」は、「<match concept.accelerometer.analyze>」ディレクティブの終了宣言である。

【0048】

40

以下に、上述の定義ファイル304にて記述される、学習データ処理部307のanalyzeモードにおけるディレクティブによって、学習データ処理部307が学習型情報処理装置104から得た分類結果であるイベントの一例を示す。なお、行頭の“#”はコメントを指し示す。

【0049】

```
# 既知の概念の中で分類できた場合
2018 01 24T23:42:42+09:00 action.recognized
{"cleaning":0.0,"blow drying":0.0,"sitting":9.84092903137207,
 "standing":0.0,"walking":0.0 ,
 "estimated label":"sitting","max score":9.84092903137207}
```

50

【 0 0 5 0 】

上記のイベントでは、「"estimated label":"sitting"」にて、推定結果が「sitting」であり、「"max score":9.84092903137207」にて、最大スコアが「9.84092903137207」という値であることが示されている。

【 0 0 5 1 】

既知の概念にあてはまるものがないと判断された場合

2018 01 24T22:46:42+09:00 concept.for train

```
{ "cleaning":0.0,"blow drying":1.9574897289276123,
  "sitting":7.8311076164245605,"standing":0.0,"walking":0.0,
  "estimated label":"sitting","max score":7.8311076164245605,
  "data":{
    "ax mean":0.143426513671875,"ay mean":0.0136016845703125,
    "az mean": 0.50174560546875,"mx mean":0.48634033203125,
    "my mean":0.26500244140625,"mz mean": 1.013330078125,
    "gx mean": 0.0011932373046875,"gy mean":0.0130523681640625,
    "gz mean":0.0047119140625
  }}

```

10

【 0 0 5 2 】

上記のイベントでは、「"estimated label":"sitting"」にて、推定結果が「sitting」であり、「"max score":7.8311076164245605」にて、最大スコアが「7.8311076164245605」という値であることが示されている。そして、最大スコアが閾値の「9」に満たないので、学習用のデータが付加されている。

20

学習用バッファ 3 0 8 は、このような推定に失敗したデータを所定の件数、蓄積する。そして、所定の件数に達した時に、trainモードにて学習型情報処理装置 1 0 4 にそれら複数のイベントを送信して、学習型情報処理装置 1 0 4 に学習を行わせる。

【 0 0 5 3 】

[情報処理装置 1 0 1 : データ処理のアーキテクチャ]

図 5 は、情報処理装置 1 0 1 が実行するデータ処理のアーキテクチャ（設計思想）を示す概念図である。

情報処理装置 1 0 1 は、3 種類のデータフローを定義し、それらデータフローをイベントのタグ名で区別する。

30

【 0 0 5 4 】

まず、マルチモーダルセンサ C 5 0 1 から生じる種々のデータは、センサレベルデータフロー C 5 0 2 に入力される。センサレベルデータフロー C 5 0 2 では、各々のイベントに「sensor.**」で始まるタグが付される。そして、センサレベルデータフロー C 5 0 2 の内部では、入力同期併合処理部 3 0 5 によって複数のイベントが 1 個のイベントに併合される、あるいは概念レベルデータフロー C 5 0 3 や行動レベルデータフロー C 5 0 4 に転送される。また定義ファイル 3 0 4 の記述によっては、センサレベルデータフロー C 5 0 2 のイベントが再びセンサレベルデータフロー C 5 0 2 にて処理されることもある。

なお、「sensor.**」の"***"（アスタリスク 2 文字）は、情報処理装置 1 0 1 が扱うイベントに付されるタグ名の文字列マッチング処理において、タグ名のフィールドセパレータである"."（ピリオド）で区切られるフィールド単位のワイルドカードを意味する。

40

【 0 0 5 5 】

次に、センサレベルデータフロー C 5 0 2 にて形成されたイベントのうち、観察対象者の A D L を認識または推定可能になったイベントは、会話音声を発する、あるいは認識または推定した A D L をデータベース 3 0 9 にログ記録する等、何らかの行動が可能になる。このような「行動可能な」イベントは「action.**」で始まるタグが付され、行動レベルデータフロー C 5 0 4 に入力される。行動レベルデータフロー C 5 0 4 では、最終的に得られた認識結果あるいは推定結果に従って、会話音声を発する、あるいは A D L をログ記録する等の、所定のアクションを実行する。このアクションがアクティベータ C 5 0 5

50

である。

【0056】

そして、センサレベルデータフローC502にて形成されたイベントのうち、そのままでは観察対象者のADLを認識できないイベントは、学習型情報処理装置104によって推定を行う必要が生じる。そこで、そのような「行動不可能な」イベントは、「concept.**」で始まるタグが付され、概念レベルデータフローC503に入力される。

【0057】

学習アルゴリズムを用いる学習型情報処理装置104は、特徴ベクトルの要素数を減らす、次元削減の効果をも有する。このため、概念レベルデータフローC503においても、センサレベルデータフローC502にて説明したと同様に、概念レベルデータフローC503のイベントが再び概念レベルデータフローC503にて処理される、ということもある。また定義ファイル304の記述によっては、イベントが概念レベルデータフローC503とセンサレベルデータフローC502を行き来する、というケースも有り得る。

概念レベルデータフローC503が出力したイベントが行動可能なイベントである場合には、センサレベルデータフローC502と同様に「action.**」で始まるタグが付され、行動レベルデータフローC504に入力される。

【0058】

概念レベルデータフローC503において、学習型情報処理装置104が推定に失敗した場合、新たな学習が必要になる。推定に失敗したイベントは、概念学習器C506を通じて、オラクルC507に問い合わせる。オラクルC507とは、正解を指し示す者であり、例えば人間のアノテータ(annotator)である。概念学習器C506は、オラクルC507と概念レベルデータフローC503との間を取り持ち、学習型情報処理装置104に対して学習を行わせる。

本発明の実施形態に係る情報処理装置101では、概念学習器C506はネットワークを通じて接続されるパソコン等の外部情報処理装置310が、その役目を担う。

【0059】

図5を見ると情報処理装置101は、マルチモーダルセンサC501から生じるイベントをセンサレベルデータフローC502と概念レベルデータフローC503との間で自由に加工を行い、最終的に行動レベルデータフローC504に与える状態にまで、観察対象者のADLを認識あるいは推定することがわかる。

また情報処理装置101は、行動レベルデータフローC504に与えるイベントを、センサレベルデータフローC502と概念レベルデータフローC503に分けることで、複雑な事象の認識や推定に対し、イベントの適切な処理を明確に指定することが可能になる。多種多様なADLに対しても、定義ファイル304に記述を追加するだけで、即座に対応することが可能になる。

【0060】

[情報処理装置101：動作の一例]

これより、図6から図9にかけて、情報処理装置101の動作の一例を示す。

図6は、マルチモーダルセンサC501からイベントを取得して、センサレベルデータフローC502でイベントの加工を行うことで観察対象者の行動認識を行い、行動レベルデータフローC504にイベントを出力する過程を示す図である。

【0061】

顔認識カメラ102が人の顔の存在を認識すると、「sensor.camera」というタグのイベントE601をセンサレベルデータフローC502に送る。センサレベルデータフローC502では、「sensor.camera」というタグのイベントE601を受け取ると、それが既に予め登録された観察対象者の顔であるか否かを、条件判断処理部306が判定する。条件判断処理部306は、登録済みの人の顔であり、且つその顔がカメラの画角の概ね中央に位置していたら、タグを「sensor.single face」に書き換えて、イベントE602を流す。

【0062】

10

20

30

40

50

観察対象者の心拍と呼吸を測定するマイクロ波センサ 6 0 3 から「sensor.microwave」というイベント E 6 0 4 がセンサレベルデータフロー C 5 0 2 に入力される。また、観察対象者の体温を非接触にて測定する非接触温度センサ 6 0 5 から「sensor.surface temp」というイベント E 6 0 6 がセンサレベルデータフロー C 5 0 2 に入力される。入力同期併合処理部 3 0 5 は、前述の「sensor.single face」イベント E 6 0 2 と、「sensor.microwave」イベント E 6 0 4 と、「sensor.surface temp」イベント E 6 0 6 を、「sensor.single face」イベント E 6 0 2 が来たタイミングで併合する。そして併合したイベント E 6 0 7 に「action.vitaldata」というタグを付して、行動レベルデータフロー C 5 0 4 に送る。

【 0 0 6 3 】

行動レベルデータフロー C 5 0 4 において「action.vitaldata」イベント E 6 0 7 を条件判断処理部 3 0 6 が受け取ると、条件判断処理部 3 0 6 は予め定義ファイル 3 0 4 にて定義された条件判断指示に従って条件判断処理を行い、アクティベータ C 5 0 5 にデータ D 6 0 8 「speakText.greeting」を流す。

例えば、「action.vitaldata」に含まれるvitaldataが正常範囲内であれば、「甲田さん、おはようございます。今日も一日楽しく過ごしましょう。」と発話する。「action.vitaldata」に含まれるvitaldataが異常値を示しているのであれば、「甲田さん、おはようございます。いつもより少し熱があるみたいです。大丈夫ですか？」と発話して、介護施設のスタッフ等にアラートを送信する。発話機能は、音声出力処理部 3 0 3、D/A変換器 2 1 0 及びスピーカ 1 0 5 によって実現される。

【 0 0 6 4 】

以上、図 6 にて説明したように、極単純な事象であれば、学習アルゴリズムを使用せずとも事象を認識して、観察対象者に対して適切な応答を行うことが可能である。

【 0 0 6 5 】

図 7 は、マルチモーダルセンサ C 5 0 1 からイベントを取得して、センサレベルデータフロー C 5 0 2 でイベントの加工を、概念レベルデータフロー C 5 0 3 にてイベントの推定を行うことで観察対象者の行動認識を行い、行動レベルデータフロー C 5 0 4 にイベントを出力する過程を示す図である。

【 0 0 6 6 】

観察対象者の行動を計測する 9 軸センサ 7 0 1 から「sensor.9dof」というイベント E 7 0 2 がセンサレベルデータフロー C 5 0 2 に入力される。また、観察対象者の発話を拾うピンマイク 7 0 3 から「sensor.mic」というイベント E 7 0 4 がセンサレベルデータフロー C 5 0 2 に入力される。入力同期併合処理部 3 0 5 は、前述の「sensor.9dof」イベント E 7 0 2 と、「sensor.mic」イベント E 7 0 4 を、「sensor.mic」イベント E 7 0 4 が来たタイミングで併合する。そして併合したイベント E 7 0 5 に「concept.9dof mic」というタグを付して、概念レベルデータフロー C 5 0 3 に送る。

【 0 0 6 7 】

概念レベルデータフロー C 5 0 3 において「concept.9dof mic」イベント E 7 0 5 を学習データ処理部 3 0 7 が受け取ると、学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 にイベント E 7 0 5 の内容、すなわち 9 軸センサのデータ及びピンマイク 7 0 3 の音声データ（特徴ベクトル、説明変数）を送信する。学習型情報処理装置 1 0 4 は情報処理装置 1 0 1 から特徴ベクトルを受信すると、学習アルゴリズムに基づく推定処理を行い、推定結果（目的変数）を情報処理装置 1 0 1 へ返信する。学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 から推定結果を受信すると、定義ファイル 3 0 4 に記述されている閾値と比較する。推定結果の値が閾値以上であり、ADLを十分に特定できた場合には、学習データ処理部 3 0 7 は推定結果のイベント E 7 0 6 に「action.recognized」というタグを付して、行動レベルデータフロー C 5 0 4 に送る。

【 0 0 6 8 】

行動レベルデータフロー C 5 0 4 において「action.recognized」イベント E 7 0 6 を条件判断処理部 3 0 6 が受け取ると、条件判断処理部 3 0 6 は予め定義ファイル 3 0 4 に

10

20

30

40

50

て定義された条件判断指示に従って条件判断処理を行い、アクティベータ C 5 0 5 にデータ D 7 0 7 を流すと共に、A D L の推定結果 D 7 0 8 をデータベース 3 0 9 にログ記録する。

【 0 0 6 9 】

以上、図 7 にて説明したように、学習型情報処理装置 1 0 4 を使用することで事象を推定して、観察対象者に対して適切な応答を行うことが可能である。

【 0 0 7 0 】

図 8 は、マルチモーダルセンサ C 5 0 1 からイベントを取得して、センサレベルデータフロー C 5 0 2 でイベントの加工を、概念レベルデータフロー C 5 0 3 にてイベントの推定を行うことで観察対象者の行動認識を行い、概念学習器 C 5 0 6 と対話して学習処理を行う過程を示す図である。

10

【 0 0 7 1 】

観察対象者の行動を計測する 9 軸センサ 7 0 1 から「sensor.9dof」というイベント E 7 0 2 がセンサレベルデータフロー C 5 0 2 に入力される。また、観察対象者の発話を拾うピンマイク 7 0 3 から「sensor.mic」というイベント E 7 0 4 がセンサレベルデータフロー C 5 0 2 に入力される。入力同期併合処理部 3 0 5 は、前述の「sensor.9dof」イベント E 7 0 2 と、「sensor.mic」イベント E 7 0 4 を、「sensor.mic」イベント E 7 0 4 が来たタイミングで併合する。そして併合したイベント E 7 0 5 に「concept.9dof mic」というタグを付して、概念レベルデータフロー C 5 0 3 に送る。

【 0 0 7 2 】

概念レベルデータフロー C 5 0 3 において「concept.9dof mic」イベント E 7 0 5 を学習データ処理部 3 0 7 が受け取ると、学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 にイベントの内容、すなわち 9 軸センサのデータ及びピンマイク 7 0 3 の音声データ（特徴ベクトル、説明変数）を送信する。学習型情報処理装置 1 0 4 は情報処理装置 1 0 1 から特徴ベクトルを受信すると、学習アルゴリズムに基づく推定処理を行い、推定結果（目的変数）を情報処理装置 1 0 1 へ返信する。

20

【 0 0 7 3 】

学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 から推定結果を受信すると、定義ファイル 3 0 4 に記述されている閾値と比較する。推定結果の値が閾値よりも小さく、A D L を十分に特定できなかった場合には、学習データ処理部 3 0 7 は推定結果 E 8 0 1 に「concept.uncategorized」というタグを付して、概念学習器 C 5 0 6 に送る。

30

【 0 0 7 4 】

学習データ処理部 3 0 7 から「concept.uncategorized」イベントを概念学習器 C 5 0 6 が受け取ると、概念学習器 C 5 0 6 はオラクル C 5 0 7 に問い合わせを行う。そして、オラクル C 5 0 7 から「concept.uncategorized」イベントに対応する正解ラベル L 8 0 2 を取得する。正解ラベルとは、学習型情報処理装置 1 0 4 が「concept.9dof mic」イベント E 7 0 5 に対して本来返答すべき正解の内容である。具体的には、本発明の実施形態の場合では、「cleaning」、「blow drying」、「sitting」、「standing」、「walking」等の、A D L の内容である。概念学習器 C 5 0 6 は正解ラベル L 8 0 2 に「concept.for training」というタグ名を付して、学習データ処理部 3 0 7 に送る。

40

【 0 0 7 5 】

正解ラベル L 8 0 2 が、情報処理装置 1 0 1 が既に認識済みの A D L の内容であるならば、学習型情報処理装置 1 0 4 に対し、既知の概念に対する再学習（既知概念拡張）を行う。正解ラベル L 8 0 2 が、情報処理装置 1 0 1 が未だ認識していない A D L の内容であるならば、学習型情報処理装置 1 0 4 に対し、未知の概念を獲得する新規の学習（未知概念獲得）を行う。その際、定義ファイル 3 0 4 に新規の概念を追加する更新処理が必要になる。

【 0 0 7 6 】

以上、図 8 にて説明したように、学習型情報処理装置 1 0 4 を使用することで事象を推定できなかった場合に、適切な再学習あるいは新規の学習を行うことが可能である。

50

【 0 0 7 7 】

図 9 は、マルチモーダルセンサ C 5 0 1 からイベントを取得して、センサレベルデータフロー C 5 0 2 における認識結果と、概念レベルデータフロー C 5 0 3 における推定結果とを組み合わせ、複数の A D L を識別し、多彩な応答を行う過程を示す図である。図 6 と図 7 を組み合わせた内容と概ね等価である。

【 0 0 7 8 】

顔認識カメラ 1 0 2 が人の顔の存在を認識すると、「sensor.camera」というタグのイベント E 6 0 1 をセンサレベルデータフロー C 5 0 2 に送る。センサレベルデータフロー C 5 0 2 では、「sensor.camera」というタグのイベント E 6 0 1 を受け取ると、それが既に予め登録された観察対象者の顔であるか否か、また 2 人以上の顔の存在を認識したか否かを、条件判断処理部 3 0 6 が判定する。条件判断処理部 3 0 6 は、2 人以上の人の顔の存在を認識した場合に、タグを「action.multiple faces」E 9 0 1 に書き換えて、イベントを行動レベルデータフロー C 5 0 4 へ流す。

10

【 0 0 7 9 】

観察対象者の行動を計測する 9 軸センサから「sensor.9dof」というイベントがセンサレベルデータフロー C 5 0 2 に入力される。また、観察対象者の発話を拾うピンマイク 7 0 3 から「sensor.mic」というイベントがセンサレベルデータフロー C 5 0 2 に入力される。

入力同期併合処理部 3 0 5 は、前述の「sensor.9dof」イベント E 7 0 2 と、「sensor.mic」イベント E 7 0 4 を、「sensor.mic」イベント E 7 0 4 が来たタイミングで併合する。そして併合したイベント E 7 0 5 に「concept.9dof mic」というタグを付して、概念レベルデータフロー C 5 0 3 に送る。

20

【 0 0 8 0 】

概念レベルデータフロー C 5 0 3 において「concept.9dof mic」イベント E 7 0 5 を学習データ処理部 3 0 7 が受け取ると、学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 にイベント E 7 0 5 の内容、すなわち 9 軸センサのデータ及びピンマイク 7 0 3 の音声データ（特徴ベクトル、説明変数）を送信する。学習型情報処理装置 1 0 4 は情報処理装置 1 0 1 から特徴ベクトルを受信すると、学習アルゴリズムに基づく推定処理を行い、推定結果（目的変数）を情報処理装置 1 0 1 へ返信する。

学習データ処理部 3 0 7 は学習型情報処理装置 1 0 4 から推定結果を受信すると、定義ファイル 3 0 4 に記述されている閾値と比較する。推定結果の値が閾値以上であり、A D L を十分に特定できた場合には、学習データ処理部 3 0 7 は推定結果のイベント E 7 0 6 に「action.recognized」というタグを付して、行動レベルデータフロー C 5 0 4 に送る。

30

【 0 0 8 1 】

センサレベルデータフロー C 5 0 2 から送られてきた「action.multiple faces」イベントと、概念レベルデータフロー C 5 0 3 から送られてきた「action.recognized」イベント E 7 0 6 は、入力同期併合処理部 3 0 5 によって併合される。併合されて単一になったイベント E 9 0 2 には「action.mood」というタグが付され、条件判断処理部 3 0 6 に送られる。

40

条件判断処理部 3 0 6 は、「action.mood」イベント E 9 0 2 を受け取ると、イベント E 9 0 2 内のデータに基づいて、種々の条件判断処理を行う。そして、発話用テキストデータ D 9 0 3 を生成してアクティベータ C 5 0 5 に引き渡す、あるいは A D L の推定結果 D 7 0 8 をデータベース 3 0 9 にログ記録する等の動作を行う。

【 0 0 8 2 】

例えば、「action.multiple faces」イベントにおいて複数人の顔を認識し、「action.recognized」イベントにて「会話」が行動ラベルとして認識された場合には、「甲田さん、乙野さん、丙山さん、なんだか楽しそうにお話をしていますね。僕も仲間に入れて下さい。」と発話する。

例えば、「action.multiple faces」イベントにおいて複数人の顔を認識し、「action.

50

recognized」イベントにて「着席」が行動ラベルとして認識された場合には、「甲田さん、乙野さん、丙山さん、そういえば、丁チームは金メダルを取ったみたいですよ。」と、ニュースを所定のニュースサイト等から取得して、発話する。またこの発話と共に、同席する人達の組み合わせ及び雰囲気情報をログファイルに記録する。

【0083】

図9を見ると、入力同期併合処理部305と条件判断処理部306は、センサレベルデータフローC502、行動レベルデータフローC504の何れにも配置できる。一方、学習データ処理部307は概念レベルデータフローC503にのみ配置できる。

入力同期併合処理部305は、複数の事象を一つのイベントとして纏めて、特徴ベクトルとして取り扱う。もし、このイベントがそのまま条件判断処理部306で処理が可能であるならば、条件判断処理部306で所定の行動を実行するための判断を行う。もし、このイベントがそのままでは条件判断処理部306で処理が不可能であるならば、学習データ処理部307に与えて、学習型情報処理装置104から推定結果を得る。

【0084】

図9を見て判るように、行動レベルデータフローC504に配置されている条件判断処理部306は、その直前に前置されている入力同期併合処理部305によって、センサレベルデータフローC502から来たイベントと、概念レベルデータフローC503から来たイベントが併合されたイベントが入力される。

また、図9ではセンサレベルデータフローC502、概念レベルデータフローC503、行動レベルデータフローC504が、それぞれ1回ずつ現れているが、センサレベルデータフローC502と概念レベルデータフローC503は、図5にて説明したように、行動レベルデータフローC504の直前において、何回でも実行することができる。

すなわち、情報処理装置101は、学習型情報処理装置104の利用を最小限に抑えつつ、多彩な条件判断が可能になる。

【0085】

このように、入力同期併合処理部305と条件判断処理部306、そして学習データ処理部307を複数個組み合わせることで、多種多様なADLを認識あるいは推定することが可能になると共に、複数のADLの識別あるいは推定結果を、他のマルチモーダルセンサC501の計測結果等と組み合わせることで、きめ細かな応答を観察対象者に実施することが可能になる。

【0086】

本発明の実施形態に係る情報処理装置101では、入出力制御部301の大部分の機能にFluentdを使用した。しかし、使用するソフトウェアは必ずしもFluentdには限らない。定義ファイル304の記述によってデータの流を自由に定義することができる仕組みがあれば、Fluentdでなくとも、情報処理装置101を実現することは可能である。一例として、名前付きパイプを応用して、シェルスクリプト等を使用すること等が考えられる。

【0087】

[学習データ処理部307と概念学習器C506及びオラクルC507]

図7では、概念レベルデータフローC503及び学習データ処理部307の動作を概略的に説明した。しかし、学習データ処理部307には、データ分類モード、新規クラス学習モードの、2種類の動作モードを有する。また更に学習データ処理部307は、学習型情報処理装置104と共に定義ファイル304を更新することで、計算機負荷の低減や推定精度の向上等が可能になる。

【0088】

図10は、入出力制御部301及び学習型情報処理装置104の、データ分類モードにおける機能ブロック図である。

マルチモーダルセンサC501から生じる種々のデータは、センサレベルデータフローC502に属する第一フィルタ1001、第二フィルタ1002、第nフィルタ1003に入力される。

第一フィルタ1001は、マルチモーダルセンサC501から生じる種々のデータのうち

10

20

30

40

50

、定義ファイル304のディレクティブに記述されたセンサのデータに対し、現在日時情報と所定のタグを付して、JSON形式のイベントを生成する。

第二フィルタ1002、また第nフィルタ1003も、第一フィルタ1001と同様に、JSON形式のイベントを生成する。

なお、第一フィルタ1001、第二フィルタ1002、第nフィルタ1003には、それぞれ同じセンサのデータがJSON形式イベントに格納されるとは限らない。すなわち、第一フィルタ1001、第二フィルタ1002、第nフィルタ1003を定義する定義ファイル304のディレクティブには、同じセンサのデータが定義されているとは限らない。

【0089】

第一フィルタ1001、第二フィルタ1002、第nフィルタ1003が出力するJSON形式のイベントは、データ形式変換処理部1004によってデータフォーマットが変換された後、学習型情報処理装置104に出力される。

学習型情報処理装置104は、データ形式変換処理部1004から入力される、第一フィルタ1001、第二フィルタ1002、第nフィルタ1003が出力するイベントに由来するデータストリームに記述されているラベル毎に、第一推定処理部1005、第二推定処理部1006、第n推定処理部1007にデータストリームを投入する。

ラベルは、イベントのタグに記述されている。例えば「sitting」「standing」等の、観察対象者の動作(ADL)を示す情報である。また、このラベルは、学習型情報処理装置104の内部に存在する推定機能ブロックを一意に識別する学習型情報処理装置104の第一推定処理部1005、第二推定処理部1006、第n推定処理部1007は、予め学習されたADLに対し、入力されたデータストリームがどの程度の確率で合致するのを示す確率を出力する。

【0090】

例えば、第一推定処理部1005がラベル名「sitting」を、第二推定処理部1006がラベル名「standing」を推定する機能ブロックであるとする。

第一フィルタ1001は、第一推定処理部1005に投入するためのセンサのデータを集め、タグに「sitting」を付したイベントを出力する。第一フィルタ1001が出力するJSON形式のイベントは、データ形式変換処理部1004によってデータフォーマットが変換された後、学習型情報処理装置104に出力される。

学習型情報処理装置104は、データ形式変換処理部1004から入力されるデータストリームに記述されている「sitting」というタグを識別すると、当該データストリームを同一名称のラベルが付されている第一推定処理部1005に振り向ける。すると第一推定処理部1005は、入力されたデータストリームに対する、観察対象者が「座っている」確率を出力する。

【0091】

第二フィルタ1002は、第二推定処理部1006に投入するためのデータを集め、タグに「standing」を付したイベントを出力する。第二フィルタ1002が出力するJSON形式のイベントは、データ形式変換処理部1004によってデータフォーマットが変換された後、学習型情報処理装置104に出力される。

学習型情報処理装置104は、データ形式変換処理部1004から入力されるデータストリームに記述されている「standing」というタグを識別すると、当該データストリームを同一名称のラベルが付されている第二推定処理部1006に振り向ける。すると第二推定処理部1006は、入力されたデータストリームに対する、観察対象者が「立っている」確率を出力する。

【0092】

つまり、第一フィルタ1001と第一推定処理部1005は、互いに同一名称のタグとラベル「sitting」で紐付いている。第二フィルタ1002と第二推定処理部1006は、互いに同一名称のタグとラベル「standing」で紐付いている。以下同様に、あるフィルタとある推定処理部は、共通する名称のタグとラベルによって、互いに1対1の対応関係

10

20

30

40

50

で紐付いている。

すなわち、センサレベルデータフロー C 5 0 2 に属するフィルタのうち、学習型情報処理装置 1 0 4 へイベントを出力するために設けられた第一フィルタ 1 0 0 1、第二フィルタ 1 0 0 2、第 n フィルタ 1 0 0 3 は、マルチモーダルセンサ C 5 0 1 から生じる種々のデータから、それぞれのタグに等しい名称のラベルに紐付く、観察対象者の A D L を推定する推定処理部に与えるイベントを出力する。

【 0 0 9 3 】

ラベル「sitting」が付されている第一推定処理部 1 0 0 5 は、入力されたデータストリーム（特徴ベクトル）に所定の演算処理を施し、観察対象者の A D L 「座っている」の確率を出力する。

10

ラベル「standing」が付されている第二推定処理部 1 0 0 6 は、入力されたデータストリーム（特徴ベクトル）に所定の演算処理を施し、観察対象者の A D L 「立っている」の確率を出力する。

以下同様に、第 n 推定処理部 1 0 0 7 まで、観察対象者の様々な A D L における確率を出力する。

これらの推定処理部が出力した A D L の確率は、データ形式逆変換処理部 1 0 0 8 に入力される。

データ形式逆変換処理部 1 0 0 8 は、第一推定処理部 1 0 0 5、第二推定処理部 1 0 0 6、第 n 推定処理部 1 0 0 7、がそれぞれ出力した A D L の確率を、ラベルをタグに変換して、JSON形式のイベントに変換する。そして、それらイベントは、ラベル特定処理部 1 0 0 9 に入力される。

20

【 0 0 9 4 】

ラベル特定処理部 1 0 0 9 は、内部に推定値閾値 1 0 1 0 とコンパレータ 1 0 1 1 を内蔵する。そして、データ形式逆変換処理部 1 0 0 8 から得られる複数個のイベントから、A D L の確率を取り出す。そして、それら A D L の確率をソート（sort：並べ替え）を行った後、最大値を示す A D L の確率に対して、コンパレータ 1 0 1 1 が推定値閾値 1 0 1 0 と比較する。

もし、ある一つの A D L の確率が推定値閾値 1 0 1 0 以上であれば、観察対象者は当該ラベルに定義される A D L を行ったと推定できる。この場合、ラベル特定処理部 1 0 0 9 は当該ラベルをタグに含み、正解の確率をデータに含むイベントを出力する。この時、当該イベントは学習用バッファ 3 0 8 には出力されない。

30

【 0 0 9 5 】

もし、全ての A D L の確率が推定値閾値 1 0 1 0 未満であれば、観察対象者がどのような A D L を行ったのかが推定できない。つまり、第一推定処理部 1 0 0 5 から第 n 推定処理部 1 0 0 7 の、全ての推定処理部が、観察対象者の A D L を推定できなかった。このことから、観察対象者の A D L は、第一推定処理部 1 0 0 5 から第 n 推定処理部 1 0 0 7 の、全ての推定処理部に定義されていない、未知の A D L を実行したものと推察される。

この場合、ラベル特定処理部 1 0 0 9 は「unknown」等の、ラベルを特定できなかったことを意味するタグを含み、マルチモーダルセンサ C 5 0 1 の全てのデータを含むイベントを出力する。この時、当該イベントは学習用バッファ 3 0 8 に出力され、記憶される。

40

【 0 0 9 6 】

図 1 1 は、入出力制御部 3 0 1 及び学習型情報処理装置 1 0 4 の、新規クラス学習モードにおける機能ブロック図である。

学習用バッファ 3 0 8 には、A D L が不明のイベントが蓄積されている。これらイベントは所定のタイミングで、概念学習用データ供給処理部 1 1 0 1 を通じて、概念学習器 C 5 0 6 へ出力される。概念学習用データ供給処理部 1 1 0 1 は、学習用バッファ 3 0 8 に蓄積されたイベントを、概念学習器 C 5 0 6 へ出力する。

【 0 0 9 7 】

概念学習器 C 5 0 6 は、概念学習用データ供給処理部 1 1 0 1 からイベントを受信すると、オラクル C 5 0 7 に問い合わせ、新規のラベルを取得する。そして、学習型情報処

50

理装置 104 に新しい推定処理部を作成する。

図 12A は、概念学習器 C506 が情報処理装置 101 とオラクル C507 と学習型情報処理装置 104 に対して実行するデータ処理の流れを示す概略図である。

まず、概念学習器 C506 は、情報処理装置 101 の概念学習用データ供給処理部 1101 からイベントを受信すると、イベントを所定のフォーマットに変換した後、オラクル C507 に送信することで、オラクル C507 に当該イベントにおけるラベルを問い合わせる。オラクル C507 は、概念学習器 C506 から受信したデータに基づき、観察対象者の A D L の正解を示すラベルを決定する。そして、ラベルを概念学習器 C506 へ送信する。

概念学習器 C506 は、オラクル C507 から受信したラベルを学習型情報処理装置 104 に送信して、新規の推定処理部の作成を依頼する。学習型情報処理装置 104 は、概念学習器 C506 からラベルを受信すると、当該ラベルに紐づく第 p 推定処理部 1201 を作成する。

【0098】

概念学習器 C506 は、fluentd の外部に設けられるデータ処理プログラムである。概念学習器 C506 は情報処理装置 101 の内部に設けてもよいし、また外部の情報処理装置にて稼働させてもよい。概念学習器 C506 は、情報処理装置 101 とオラクル C507 と学習型情報処理装置 104 とのデータ等の送受信を行うことで、情報処理装置 101 と学習型情報処理装置 104 に、共通の名称で紐づく機能ブロックを生成する。すなわち概念学習器 C506 は、情報処理装置 101 におけるタグ名にて特定されるフィルタと、学習型情報処理装置 104 におけるラベル名にて特定される推定処理部を生成する。

【0099】

概念レベルデータフロー C503 の説明で触れたが、オラクル C507 とは、概念学習器 C506 による観察対象者の A D L の問い合わせに対し、正解を指し示す者または装置である。例えば、概念学習器 C506 が出力したマルチモーダルセンサ C501 のデータに基づいて観察対象者の A D L を推定する、情報処理装置 101 より上位に位置する他の学習型情報処理装置がオラクル C507 に該当する。また、概念学習器 C506 が出力したマルチモーダルセンサ C501 のデータに付されている日時情報に基づいて、監視担当者または画像認識装置が観察対象者の監視カメラ映像を見ることで、観察対象者の A D L を判断する、という作業もオラクル C507 に該当する。

【0100】

図 12B は、概念学習器 C506 が情報処理装置 101 と学習型情報処理装置 104 に対して実行するデータ処理の流れを示す概略図である。

図 12A において、概念学習器 C506 が学習型情報処理装置 104 に第 p 推定処理部 1201 を作成した。次に、概念学習器 C506 は情報処理装置 101 の概念学習用データ供給処理部 1101 から再びイベントを受信する。そして、イベントを所定のフォーマットに変換して学習用データを形成した後、学習型情報処理装置 104 に学習用データを送信することで、第 p 推定処理部 1201 に対する学習処理を実行する。すなわち、概念学習器 C506 は第 p 推定処理部 1201 に対応する第 p 学習処理部 1202 に、学習用データを提供する。

【0101】

学習型情報処理装置 104 は概念学習器 C506 から受信したラベルと学習用データを用いて、第 p 学習処理部 1202 に学習処理を行う。すると、学習型情報処理装置 104 は、マルチモーダルセンサ C501 が出力するデータ（特徴ベクトル）に対し、第 p 推定処理部 1201 を、オラクル C507 が示したラベルに該当する A D L を最大事後確率として出力する関数とする学習処理を行う。

【0102】

図 12C は、概念学習器 C506 が情報処理装置 101 に対して実行するデータ処理の流れを示す概略図である。

図 12B において、概念学習器 C506 が学習型情報処理装置 104 の第 p 学習処理部

10

20

30

40

50

1202に対し、学習処理を実行した。次に概念学習器C506は、情報処理装置101のセンサレベルデータフローC502に属する新たなフィルタとして、第p推定処理部1201のラベルと同名のタグを有する第pフィルタ1203のディレクティブを作成して、定義ファイル304に登録する。

こうして概念学習器C506は、オラクルC507から新たなADLを定義するラベルを得て、情報処理装置101におけるタグ名にて特定される第pフィルタ1203と、学習型情報処理装置104におけるラベル名にて特定される第p推定処理部1201を生成することができる。

なお、図10から図12にかけて説明した実施形態に係る学習型情報処理装置104では、推定処理部をADL(クラス)毎に作っている。これは、例えばナイーブベイズ等の、2クラス分類の組み合わせで多クラス分類をする方法を模式化している。もし、ランダムフォレストのような決定木法を採用した場合は、そのまま多クラス分類が可能である。

10

【0103】

[定義ファイル304の更新と、学習型情報処理装置104の最適化処理]

定義ファイル304には、センサレベルデータフローC502に属する第一フィルタ1001、第二フィルタ1002、第nフィルタ1003、第pフィルタ1203のディレクティブが記述されている。これらフィルタのディレクティブには、マルチモーダルセンサC501が出力するデータのうち、タグに対応するラベルによって紐付けされる、学習型情報処理装置104の推定処理部に、特徴ベクトルを与える役割を有する。本発明の実施形態に係る情報処理装置101における特徴ベクトルとは、マルチモーダルセンサC501に属する様々なセンサが出力する、種々のデータ群である。

20

【0104】

学習アルゴリズムにおいて、特徴ベクトルを構成する要素には寄与度というものが存在する。例えば、観察対象者に対して種々のセンサを設けて、それらセンサのデータを情報処理装置101が受信して処理をする状況を考える。

例えば、観察対象者が「立っている」というADLに対し、あるセンサの信号はその値によって観察対象者が立っているか否かを推定する際に大きな役割を有する。その一方で、別のあるセンサの信号はその値の如何にかかわらず、観察対象者が立っているか否かを推定する際に全く役に立たない。

30

【0105】

また、観察対象者が「座っている」というADLに対し、あるセンサの信号はその値によって観察対象者が座っているか否かを推定する際に大きな役割を有する。その一方で、別のあるセンサの信号はその値の如何にかかわらず、観察対象者が座っているか否かを推定する際に全く役に立たない。

すなわち、あるADLの確率を推定する推定処理部は、以上に説明したような特徴ベクトルの要素によって出力される確率に対する寄与度が異なる。

【0106】

推定処理は、特徴ベクトルの要素数が少なければ少ないほど、演算処理の負荷が軽くなる。したがって、特徴ベクトルの要素のうち、寄与率に全く貢献しない要素は予め定義ファイル304に記述されているフィルタのディレクティブにて除外しておけば、学習型情報処理装置104の演算負荷が軽くなり、システム全体のパフォーマンスが向上する。

40

【0107】

図13は、情報処理装置101における、定義ファイル304の更新処理を示すブロック図である。この処理は、例えば情報処理装置101が通常の処理を行う必要がない、観察対象者が就寝している深夜等の時間帯において、システムメンテナンス処理の一環として実行される。図13では一例として、図12で説明した第pフィルタ1203と第p推定処理部1201を更新するものとする。

フィルタ抜粋処理部1301は、定義ファイル304を読み込むと、センサレベルデータフローC502に属する第pフィルタ1203のディレクティブを抜粋する。そして、

50

抜粋した第 p フィルタ 1 2 0 3 のディレクティブのうち、第 p 推定処理部 1 2 0 1 を特定するラベルと同名のタグと、センサのエントリを、重要度算出部 1 3 0 2 へ出力する。

【 0 1 0 8 】

重要度算出部 1 3 0 2 は、ランダムフォレスト等の周知のアルゴリズムを用いて、学習用バッファ 3 0 8 に記憶されているデータを読み出し、第 p 推定処理部 1 2 0 1 にデータ（特徴ベクトル）を投入することで、特徴ベクトルに属するセンサのデータの重要度を算出する。

重要度算出部 1 3 0 2 が算出したセンサのデータ、すなわち特徴ベクトルの要素の重要度は、特徴圧縮処理部 1 3 0 3 に入力される。特徴圧縮処理部 1 3 0 3 は、重要度算出部 1 3 0 2 が算出した重要度に従って特徴ベクトルの要素に順位を付す。更に、削除しても推定処理の精度が所定の閾値より低下しない程度まで、特徴ベクトルの要素、すなわち寄与率の低いセンサのデータを、センサのエントリから削除する。そして、特徴圧縮処理部 1 3 0 3 は圧縮したセンサのエントリをラベルと共に概念学習器 C 5 0 6 に送信する。

概念学習器 C 5 0 6 は、前述の図 1 2 において説明したように、新たな推定処理部を生成し、推定処理部に対応する学習処理部に学習処理を施す。そして、圧縮したセンサのエントリとラベル、すなわちタグに基づいて、新たなフィルタのエントリを作成し、定義ファイルに追記する。

【 0 1 0 9 】

以下に、定義ファイル 3 0 4 にて記述される、定義ファイル 3 0 4 の更新処理が実行される前のフィルタの指示（ディレクティブ（directive））の一例を示す。

【 0 1 1 0 】

```
<match concept.9dof and mic>
  @type aol classifier
  host 127.0.0.1
  port 9199
  client api 'classifier'
train analyze analyze
threshold 9.0
target class sitting
num keys mfcc p,mfcc 1,mfcc 2,mfcc 3,mfcc 4,mfcc 5,
mfcc 6,mfcc 7,mfcc 8,mfcc 9,mfcc 10,mfcc 11,mfcc 12,
rms mean,rms var,zcr,ax mean,ay mean,az mean,mx mean,
my mean,mz mean,gx mean,gy mean,gz mean,ax var,ay var,
az var,mx var,my var,mz var,gx var,gy var,gz var
  tag action.recognized.9dof and mic
</match>
```

【 0 1 1 1 】

以下に、定義ファイル 3 0 4 にて記述される、定義ファイル 3 0 4 の更新処理が実行された後のフィルタの指示（ディレクティブ（directive））の一例を示す。

【 0 1 1 2 】

```
<match concept.9dof and mic>
  @type aol classifier
  host 127.0.0.1
  port 9199
  client api 'classifier'
train analyze analyze
threshold 9.0
target class sitting
num keys az mean,ay mean,my mean
  tag action.recognized.9dof and mic
```

10

20

30

40

50

</match>

【 0 1 1 3 】

1行目の「<match concept.9dof and mic.train>」は、「concept.9dof and mic」というタグを持つデータに対して、「<match >」から始まり「</match>」で終端する中に記述されているパラメータを利用して処理を行うことを意味する。「concept.9dof and mic」には、「9dof」というセンサと「mic」から得られる特徴量をマージさせた、JSON形式のイベントが流れてくる。2行目の「@type aol classifier」で、入力データに対して「aol classifier」の処理を適用することを指定する8行目の「target class」では、どのクラスを分類するための分類器なのかを指定する。今回の例は「sitting」としている。

10

9行目の「num keys」で、分類器が使う特徴量データのkeyを指定する。

定義ファイル304の更新処理によって、9行目の「num keys」に列挙される特徴ベクトルの要素数が、大幅に減少していることがわかる。

【 0 1 1 4 】

[シリアライザ1402による対話型シナリオエンジンの実現]

図5にて説明したように、情報処理装置101は、マルチモーダルセンサC501から発生したデータに対し、観察対象者がどのようなADLを行っているのかをセンサレベルデータフローC502にて特定し、あるいは概念レベルデータフローC503にて推定して、それらADLに対応する反応、すなわちアクションを行動レベルデータフローC504で決定し、決定したアクションをアクティベータC505にて実現する。

20

【 0 1 1 5 】

前述のように、情報処理装置101におけるアクティベータC505は、音声出力処理部303、D/A変換器210及びスピーカ105によって実現される発話機能を含む。例えば、前述の発話の例である「甲田さん、おはようございます。今日も一日楽しく過ごしましょう。」という発話は、発話開始から完了までおよそ4～5秒程度の時間を要する。

【 0 1 1 6 】

マルチモーダルセンサC501から発生したデータによっては、同時刻に複数のADLが確認でき、それによって複数のアクションが同時に発生するケースがありうる。その場合、何も制御を行わずに複数のアクションを同時に音声出力処理部303に与えると、複数の発話が同時に発生してしまう。

30

そこで、アクションに優先度を付与し、発生したアクションを一旦キュー（バッファメモリ）に格納してから優先度でソートし、先に実行されているアクションが終了するまで次のアクションの実行を待つ制御を行うことで、観察対象者との自然な対話が可能になる。

【 0 1 1 7 】

図14は、本発明の変形例に係る情報処理装置1401のソフトウェア機能を示すブロック図である。

図14に示す情報処理装置1401の、図3に示す情報処理装置101との相違点は、情報処理装置1401にはシリアライザ1402があることのみである。シリアライザ1402以外の機能ブロックは、図3に示される機能ブロックと同一であるので、説明を省略する。

40

【 0 1 1 8 】

図15は、シリアライザ1402の機能ブロック図である。

シリアライザ1402は、アクション選択処理部1501とキュー1502よりなる。

アクション選択処理部1501は、入力同期併合処理部305、条件判断処理部306及び学習データ処理部307からアクションに係るイベントを受信すると、それらイベントを一旦キュー1502に記憶する。そして、イベントのタグに付されている優先度に基づいてソートを行い、優先度の高いイベントから順番に処理を行う。また、優先度が付されていない、音声出力処理部303に出力しないイベントは、データベース309に出力

50

する。

なお、本発明の実施形態では明示していないが、アクションは上記の発話とログ記録に限られない。例えば、ブザーを鳴動させる、LEDを発光させる、LCD等の表示装置に所定のメッセージ等を表示させる他、本発明をロボットに組み込む場合は、ロボットの挙動自体がアクションとなる。

【0119】

[シナリオの優先度]

本発明の変形例に係る情報処理装置1401は、アクションを決定する行動レベルデータフローC504において、シナリオを優先度に応じてグループ分けを行う。

以下に、シナリオの優先度を列挙する。

10

【0120】

・異常対応シナリオ(優先度4)

観察対象者が転倒、発作、梗塞を起こした等、観察対象者が緊急性を要する状態を検知し、アラームを発するのためのシナリオである。

優先度4レベルのシナリオは、以下に記す下位の優先度のシナリオの実行を中断してでも、最優先で実行する。

・健康チェックシナリオ(優先度3)

観察対象者の介護カルテ等に記録を残す、観察対象者のバイタルデータ取得用のシナリオである。

優先度3レベルのシナリオは、以下に記す下位の優先度のシナリオの実行に優先して実行する。

20

・リラクゼーションシナリオ(優先度2)

観察対象者に対する挨拶、ゲーム、お歌等、観察対象者をリラックスさせるためのシナリオである。

優先度2レベルのシナリオは、以下に記す下位の優先度のシナリオの実行に優先して実行する。

・応答シナリオ(優先度1)

観察対象者の問いかけに対して応答する、受動的応答のシナリオである。

優先度1レベルのシナリオは、観察対象者の問いかけ、情報処理装置1401の起動時等のシチュエーションにおいて、かつ、上位のシナリオが実行されていない時に実行する。

30

【0121】

[イベントに対するシナリオの優先度の付与]

本発明の変形例に係る情報処理装置1401は、アクションを決定する行動レベルデータフローC504において、シナリオを優先度に応じてグループ分けを行い、所定の条件に合致するイベントに対してシナリオの優先度をタグ名にて付与する。

以下に、定義ファイル304にて記述される条件判断処理部306の指示(ディレクティブ(directive))の一例を示す。

【0122】

```
<match action.greeting>
  @type condition checker
  tag1 scenario.priority 3.vital check
  tag2 scenario.priority 1.speak text.greet back
  tag else none
  renew record true
  condition1 record['nameNum'] == 1
  condition2 record['nameNum'] > 1
  <record1>
    text ${record['names'] + record['word']}
  </record1>
```

40

50

<record2>

// 今回の例では、vital checkではデータの中身は必要なし

</record2>

</match>

【 0 1 2 3 】

上記のディレクティブの例は、挨拶アクションが認識された時 (action.greeting) に、ロボットの前で検出された人数が 1 人だけの場合はバイタルチェックシナリオ (優先度 3) を、ロボットの前で検出された人が 2 人以上の場合は名前を呼んで挨拶する (優先度 1) という設定ファイルの記述の例である。条件に合致した (Matchした) データが「 condition1」で指定されている条件を満たしている場合には、タグ名を「 tag1」で指定されている「 scenario.priority 3.vital check」に書き換えて送信する。同様に、「 condition2」で指定されている条件を満たしている場合には、「 tag2」で指定されている「 scenario.priority 1.speak text.greet back」に書き換える。「 scenario.priority *」の「 *」に該当している部分は、シナリオデータの優先度を示している。その後にある「 vital check」や「 speak text」は、実行するシナリオの種類に対応している。

10

【 0 1 2 4 】

[シリアライザ 1 4 0 2 の動作]

図 1 6 は、シリアライザ 1 4 0 2 の動作の流れを示すフローチャートである。

処理を開始すると (S 1 6 0 1)、アクション選択処理部 1 5 0 1 はイベントのタグに付されている優先度を検証する (S 1 6 0 2)。

20

ステップ S 1 6 0 2 から S 1 6 0 8 までは、プログラミング言語における switch - case 文等による条件分岐処理である。

優先度が 4 である場合 (S 1 6 0 2 の「優先度 = 4」)、アクション選択処理部 1 5 0 1 は異常対応シナリオ処理を実行して (S 1 6 0 3)、一連の処理を終了する (S 1 6 0 4)。

優先度が 3 である場合 (S 1 6 0 2 の「優先度 = 3」)、アクション選択処理部 1 5 0 1 は健康チェックシナリオ処理を実行して (S 1 6 0 5)、一連の処理を終了する (S 1 6 0 4)。

優先度が 2 である場合 (S 1 6 0 2 の「優先度 = 2」)、アクション選択処理部 1 5 0 1 はリラクゼーションシナリオ処理を実行して (S 1 6 0 6)、一連の処理を終了する (S 1 6 0 4)。

30

優先度が 1 である場合 (S 1 6 0 2 の「優先度 = 1」)、アクション選択処理部 1 5 0 1 は応答シナリオ処理を実行して (S 1 6 0 7)、一連の処理を終了する (S 1 6 0 4)。

優先度が付されていない場合 (S 1 6 0 2 の「優先度なし」)、アクション選択処理部 1 5 0 1 はログ記録処理を実行して (S 1 6 0 8)、一連の処理を終了する (S 1 6 0 4)。

何れの処理においても、ステップ S 1 6 0 4 から再びステップ S 1 6 0 1 に戻り、再度処理が繰り返される。

【 0 1 2 5 】

40

図 1 7 は、異常対応シナリオ処理の動作の流れを示すフローチャートである。図 1 6 のステップ S 1 6 0 3 の詳細である。

処理を開始すると (S 1 7 0 1)、アクション選択処理部 1 5 0 1 はキュー 1 5 0 2 に格納されている優先度 3 以下のイベントを全て削除 (クリア) する (S 1 7 0 2)。次に、音声出力処理部 3 0 3 にて実行されている実行中のアクションを中断して (S 1 7 0 3)、直ちに現在のアクション、すなわち優先度 4 のアクションを実行して (S 1 7 0 4)、一連の処理を終了する (S 1 7 0 5)。

【 0 1 2 6 】

図 1 8 は、健康チェックシナリオ処理、リラクゼーションシナリオ処理、応答シナリオ処理の動作の流れを示すフローチャートである。図 1 6 のステップ S 1 6 0 5、S 1 6 0

50

6 及び S 1 6 0 7 の詳細である。

処理を開始すると (S 1 8 0 1)、アクション選択処理部 1 5 0 1 はキュー 1 5 0 2 が空であるか否かを確認する (S 1 8 0 2)。

キュー 1 5 0 2 が空であれば (S 1 8 0 2 の Y E S)、アクション選択処理部 1 5 0 1 は行動レベルデータフロー C 5 0 4 から受信したイベントをキュー 1 5 0 2 に格納して (S 1 8 0 3)、一連の処理を終了する (S 1 8 0 4)。

キュー 1 5 0 2 が空でなければ (S 1 8 0 2 の N O)、アクション選択処理部 1 5 0 1 は行動レベルデータフロー C 5 0 4 から受信したイベントをキュー 1 5 0 2 に格納する (S 1 8 0 5)。その後、キュー 1 5 0 2 の中身を優先度でソートして (S 1 8 0 6)、一連の処理を終了する (S 1 8 0 4)。

【 0 1 2 7 】

ステップ S 1 8 0 6 におけるキュー 1 5 0 2 のソート処理は、キュー 1 5 0 2 に最初のイベントが入った時点から所定の時間、例えば 0. 5 秒等経過後に実行するようにアクション選択処理部 1 5 0 1 の機能を構成してもよい。つまり、キュー 1 5 0 2 にある程度のレコード数のイベントが蓄積された時点でソート処理を実行することで、無駄なソート処理の繰り返しを抑制する。

【 0 1 2 8 】

なお、図 1 8 では説明を簡単にするために、優先度 3、2 及び 1 の処理を同一のフローチャートで説明したが、勿論、異なる処理にしてもよい。例えば、優先度 1 のイベントに対する応答シナリオ処理 S 1 6 0 7 ついては、キュー 1 5 0 2 に優先度 3 のイベントが格納されていたなら、優先度 1 のイベントをキュー 1 5 0 2 に格納せずに破棄し、実行をしない、という処理にしてもよい。

【 0 1 2 9 】

図 1 9 は、キュー 1 5 0 2 の状態を説明するための概略図である。

今、(a) に示すように、キュー 1 5 0 2 には 5 個のレコードのイベントが格納されているとする。キュー 1 5 0 2 には、イベントに付されている時刻順にイベントが格納されているが、それらイベントに付されている優先度はばらばらである。

次に、図 1 8 のステップ S 1 8 0 6 において、アクション選択処理部 1 5 0 1 がキュー 1 5 0 2 をソートすると、(b) に示すように、キュー 1 5 0 2 のレコードは優先度が優先してソートされる。この結果、キュー 1 5 0 2 の最下部レコードの優先度が最も高い「 3 」のレコードになり、レコードの順番が上に行くに従って、優先度が低い数のレコードが積み上がっている。

【 0 1 3 0 】

本発明の各実施形態においては、情報処理装置 1 0 1 を開示した。

情報処理装置 1 0 1 は、種々のセンサ (マルチモーダルセンサ C 5 0 1) を利用して、多種多様な A D L の識別あるいは推定を行う。

入力同期併合処理部 3 0 5 は、複数のイベントを受け入れて、各々のイベントの内容を単一のイベントに併合して出力する。その際、入力同期併合処理部 3 0 5 は、定義ファイル 3 0 4 に記述することで指定されたタイミングで、イベントを出力する。

【 0 1 3 1 】

学習データ処理部 3 0 7 は、入力されたイベントの内容を学習型情報処理装置 1 0 4 に特徴ベクトルとして出力し、学習型情報処理装置 1 0 4 から目的変数を受け取り、イベントの形式に変換して出力する。

条件判断処理部 3 0 6 は、単一のイベントを受け入れて、イベントに含まれているデータに対し、所定の条件判断を行うことで、条件に応じて異なるイベントを出力する。

情報処理装置 1 0 1 は、入力同期併合処理部 3 0 5、学習データ処理部 3 0 7、そして条件判断処理部 3 0 6 を定義ファイル 3 0 4 にて複数回使用することが可能であり、これらを組み合わせることで、複雑なデータフローを容易に形成し、多種多様な A D L の識別あるいは推定を行うことができる。

【 0 1 3 2 】

10

20

30

40

50

以上、本発明の実施形態について説明したが、本発明は上記実施形態に限定されるものではなく、請求の範囲に記載した本発明の要旨を逸脱しない限りにおいて、他の変形例、応用例を含む。

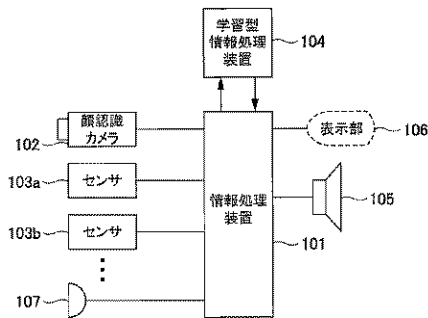
【0133】

101 情報処理装置、102 顔認識カメラ、103a、103b センサ、104 学習型情報処理装置、105 スピーカ、106 表示部、107 マイク、201 バス、202 CPU、203 ROM、204 RAM、205 不揮発性ストレージ、206 リアルタイムクロック、207 NIC、208 シリアルI/F、209 A/D変換器、210 D/A変換器、211 操作部、301 入出力制御部、302 入出力整形処理部、303 音声出力処理部、304 定義ファイル、305 入力同期併合処理部、306 条件判断処理部、307 学習データ処理部、308 学習用バッファ、309 データベース、310 外部情報処理装置、603 マイクロ波センサ、605 非接触温度センサ、701 9軸センサ、703 ピンマイク、1001 第一フィルタ、1002 第二フィルタ、1003 第nフィルタ、1004 データ形式変換処理部、1005 第一推定処理部、1006 第二推定処理部、1007 第n推定処理部、1008 データ形式逆変換処理部、1009 ラベル特定処理部、1010 推定値閾値、1011 コンパレータ、1101 概念学習用データ供給処理部、1201 第p推定処理部、1202 第p学習処理部、1203 第pフィルタ、1301 フィルタ抜粋処理部、1302 重要度算出部、1303 特徴圧縮処理部、1401 情報処理装置、1402 シリアライザ、1501 アクション選択処理部、1502 キュー

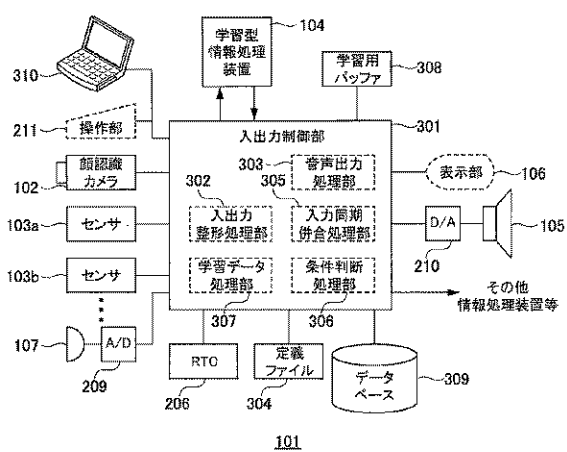
10

20

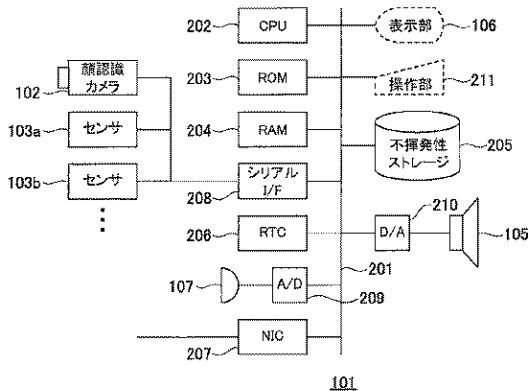
【図1】



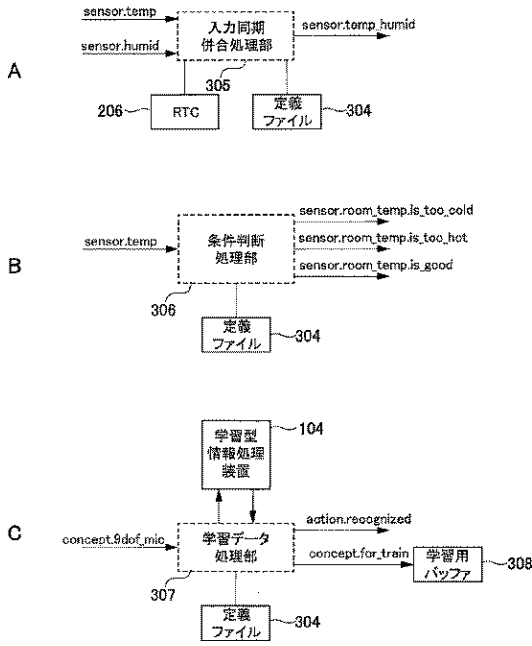
【図3】



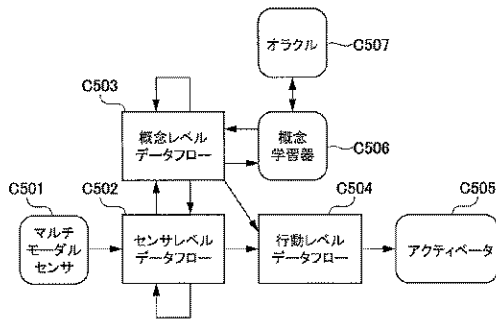
【図2】



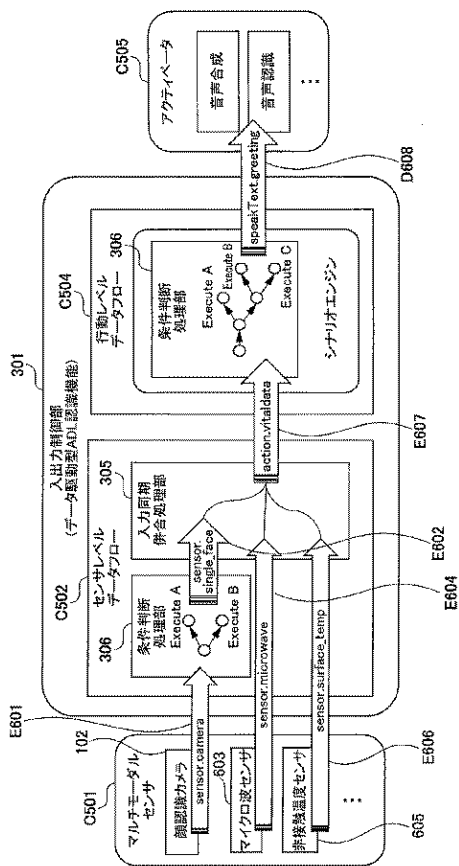
【図4】



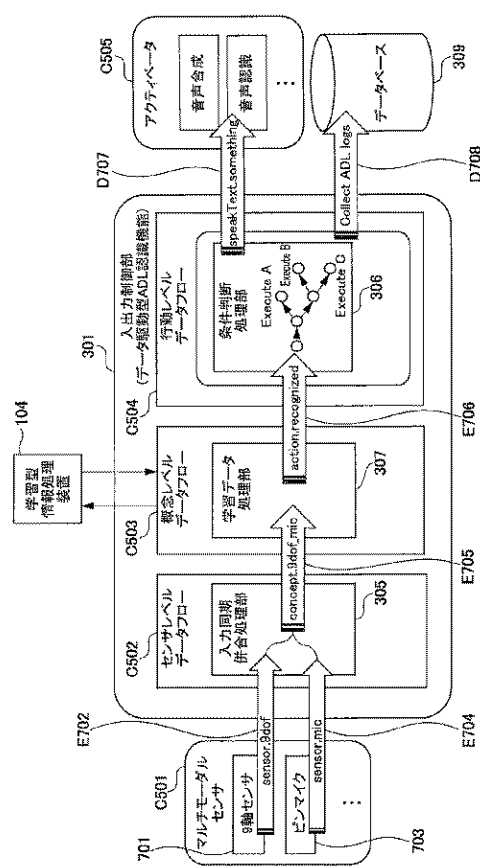
【図5】



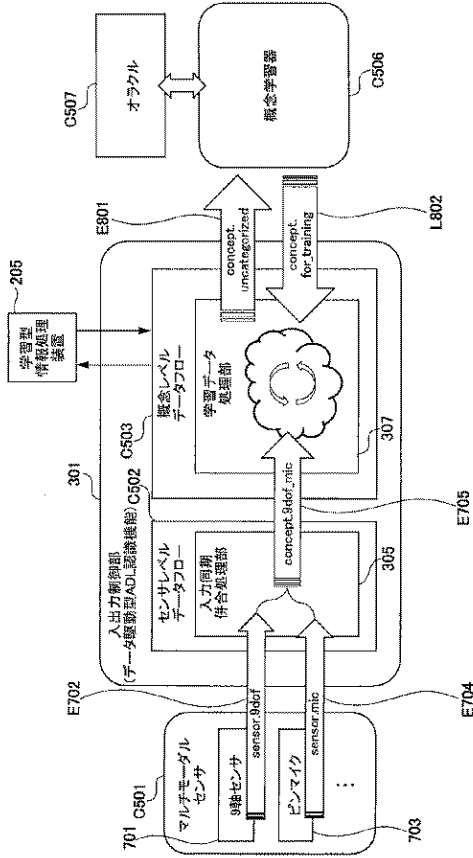
【図6】



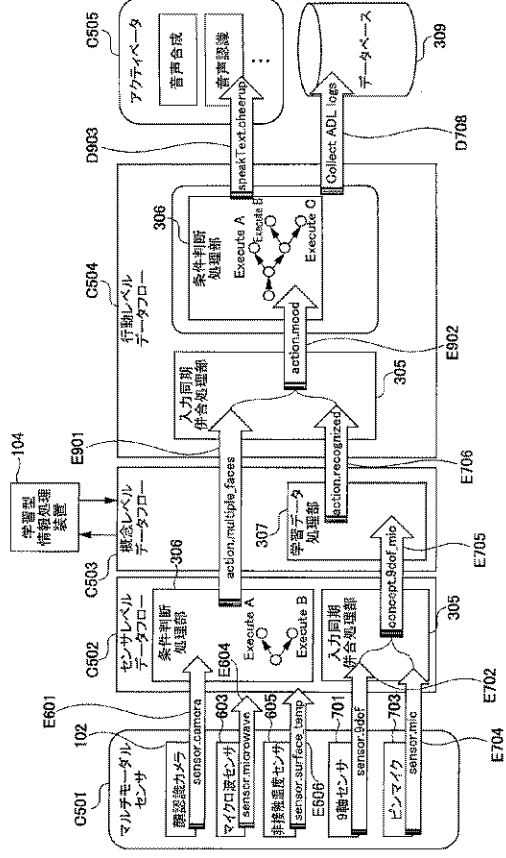
【図7】



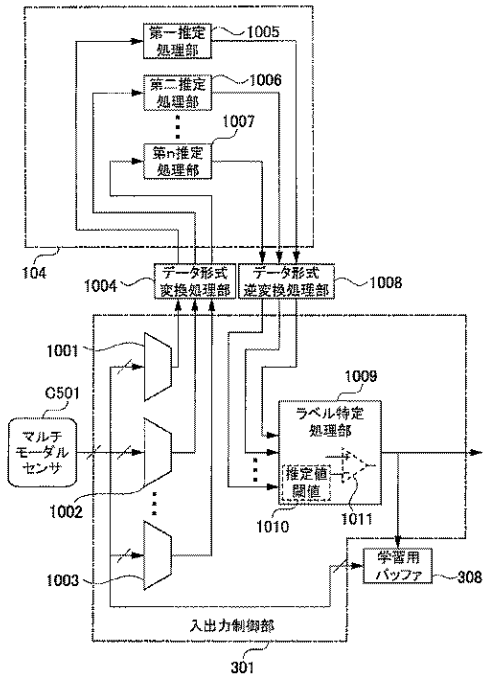
【図 8】



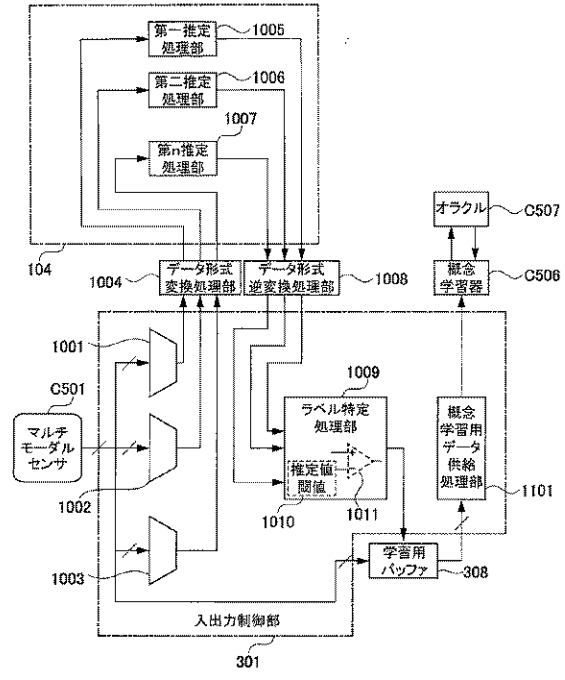
【図 9】



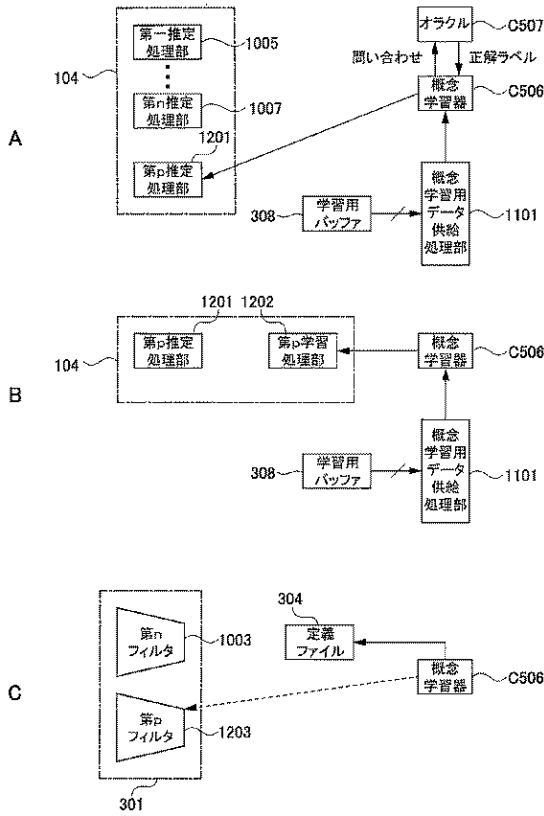
【図 10】



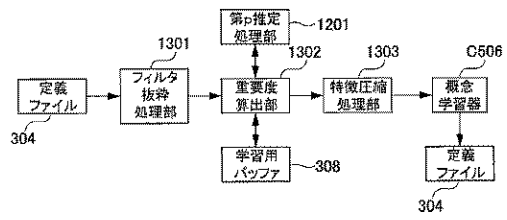
【図 11】



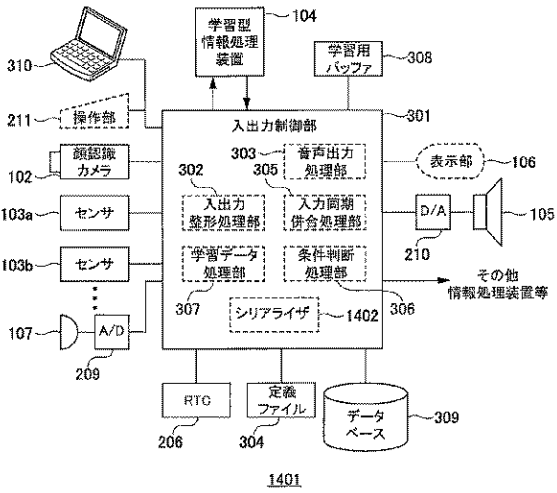
【図12】



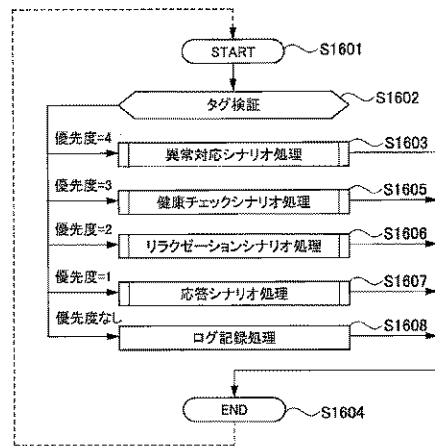
【図13】



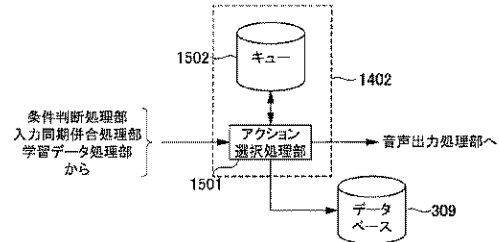
【図14】



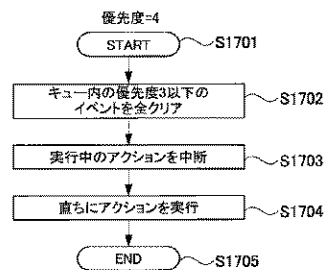
【図16】



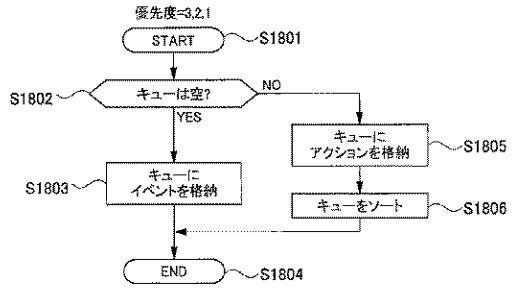
【図15】



【図17】



【図 18】



【図 19】

