

(11)特許出願公開番号

特開2010-72718

(P2010-72718A)

(43) 公開日 平成22年4月2日(2010.4.2)

(51) Int.Cl.
G06F 15/17

F 1
G06F 15/17 635A

テーマコード (参考)
5B045

審査請求 未請求 請求項の数 6 O L (全 18 頁)

(21) 出願番号 特願2008-236686 (P2008-236686)
(22) 出願日 平成20年9月16日 (2008. 9. 16)

(71) 出願人 000004226
日本電信電話株式会社
東京都千代田区大手町二丁目3番1号

(74) 代理人 100064908
弁理士 志賀 正武

(74) 代理人 100108453
弁理士 村山 靖彦

(74) 代理人 100148873
弁理士 渡辺 浩史

(72) 発明者 南 裕也
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内

(72) 発明者 荒川 豊
東京都千代田区大手町二丁目3番1号 日本電信電話株式会社内

[最終頁に続く](#)

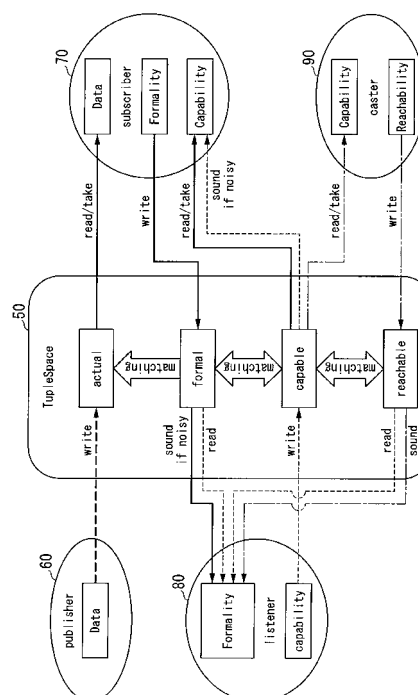
(54) 【発明の名称】 間接通信システム、間接通信方法、及びプログラム

(57) 【要約】

【課題】書き込み用Tuple (Actual) との不必要なマッチング処理を削減しマッチング処理の効率を向上させること。

【解決手段】第2のタプルを書き込む際に、第1のタプルの属性情報を当該第2のタプルの属性条件で検索して読み出し、第3のタプルの属性条件を当該第2のタプルの属性条件で検索して読み出し、この第3のタプルの登録端末に当該第2のタプルを通知し、第3のタプルを書き込む際に、第2のタプルの属性条件を当該第3のタプルの属性条件で検索して読み出し、この第2のタプルの登録端末に当該第3のタプルを通知し、第4のタプルの属性条件を当該第3のタプルの属性条件で検索して読み出し、第4のタプルの登録端末に当該第3のタプルを通知し、第4のタプルを書き込む際に、第3のタプルの属性条件を当該第4のタプルの属性条件で検索して読み出し、この第3のタプルの登録端末に当該第4のタプルを通知する。

【選択図】図4



【特許請求の範囲】**【請求項 1】**

複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタブルを読み書きすることで通信を行う間接通信システムであって、

第 1 のデータと当該第 1 のデータの属性情報とを有する第 1 のタブルを管理する装置と

、
所望するデータの属性条件を有する第 2 のタブルを管理する装置と、

所望する前記第 2 のタブルの属性条件、又は、任意の前記第 2 のタブルを管理する装置に読み出される第 2 のデータと当該第 2 のデータの属性情報を有する第 3 のタブルを管理する装置と、

第 3 のデータと前記第 3 のタブルを指定するための属性条件を有する第 4 のタブルを管理する装置と、

を備え、

前記第 1 のタブルを管理する装置は、前記第 1 のタブルを前記記憶空間に書き込み、

前記第 2 のタブルを管理する装置は、前記第 2 のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 1 のタブルの属性情報を当該第 2 のタブルの属性条件で検索し、検索された前記第 1 のタブルを読み出し、前記記憶空間に書き込まれた前記第 3 のタブルのうち属性条件が当該第 2 のタブルの属性条件と一致するものを検索し、検索された前記第 3 のタブルを読み出し、検索された前記第 3 のタブルを前記記憶空間に書き込んだ装置に当該第 2 のタブルを送信し、

前記第 3 のタブルを管理する装置は、前記第 3 のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタブルのうち属性条件が当該第 3 のタブルの属性条件と一致するものを検索し、検索された前記第 2 のタブルを読み出し、検索された前記第 2 のタブルを前記記憶空間に書き込んだ装置に当該第 3 のタブルを送信し、前記記憶空間に書き込まれた前記第 4 のタブルのうち属性条件が当該第 3 のタブルの属性条件と一致するものを検索し、検索された前記第 4 のタブルを読み出し、検索された前記第 4 のタブルを前記記憶空間に書き込んだ装置に当該第 3 のタブルを送信し、

前記第 4 のタブルを管理する装置は、前記第 4 のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 3 のタブルのうち属性条件が当該第 4 のタブルの属性条件と一致するものを検索し、検索された前記第 3 のタブルを読み出し、検索された前記第 3 のタブルを前記記憶空間に書き込んだ装置に当該第 4 のタブルを送信する、

間接通信システム。

【請求項 2】

前記第 1 のタブルを管理する装置は、前記第 1 のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタブルのうち属性条件が当該第 1 のタブルの属性情報を包含するものを検索し、検索された前記第 2 のタブルを前記記憶空間に書き込んだ装置に当該第 1 のタブルを送信することを特徴とする、請求項 1 に記載の間接通信システム。

【請求項 3】

複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタブルを読み書きすることで通信を行う間接通信システムにおいて、

第 1 のデータと当該第 1 のデータの属性情報とを有する第 1 のタブルを管理する装置が、前記第 1 のタブルを前記記憶空間に書き込むステップと、

所望するデータの属性条件を有する第 2 のタブルを管理する装置が、前記第 2 のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 1 のタブルの属性情報を当該第 2 のタブルの属性条件で検索し、検索された前記第 1 のタブルを読み出し、前記記憶空間に書き込まれた前記第 3 のタブルのうち属性条件が当該第 2 のタブルの属性条件と一致するものを検索し、検索された前記第 3 のタブルを読み出し、検索された前記

10

20

30

40

50

第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 2 のタプルを送信するステップと、

、
所望する前記第 2 のタプルの属性条件、又は、任意の前記第 2 のタプルを管理する装置に読み出される第 2 のデータと当該第 2 のデータの属性情報を有する第 3 のタプルを管理する装置が、前記第 3 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 2 のタプルを読み出し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信し、前記記憶空間に書き込まれた前記第 4 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 4 のタプルを読み出し、検索された前記第 4 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信するステップと、

第 3 のデータと前記第 3 のタプルを指定するための属性条件を有する第 4 のタプルを管理する装置が、前記第 4 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 3 のタプルのうち属性条件が当該第 4 のタプルの属性条件と一致するものを検索し、検索された前記第 3 のタプルを読み出し、検索された前記第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 4 のタプルを送信するステップと、

を含む間接通信方法。

【請求項 4】

前記第 1 のタプルを管理する装置が、前記第 1 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 1 のタプルの属性情報を包含するものを検索し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 1 のタプルを送信するステップをさらに含むことを特徴とする、請求項 3 に記載の間接通信方法。

【請求項 5】

複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタプルを読み書きすることで通信を行う間接通信システムにおいて、

第 1 のデータと当該第 1 のデータの属性情報とを有する第 1 のタプルを前記記憶空間に書き込むステップと、

所望するデータの属性条件を有する第 2 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 1 のタプルの属性情報を当該第 2 のタプルの属性条件で検索し、検索された前記第 1 のタプルを読み出し、前記記憶空間に書き込まれた前記第 3 のタプルのうち属性条件が当該第 2 のタプルの属性条件と一致するものを検索し、検索された前記第 3 のタプルを読み出し、検索された前記第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 2 のタプルを送信するステップと、

所望する前記第 2 のタプルの属性条件、又は、任意の前記第 2 のタプルを管理する装置に読み出される第 2 のデータと当該第 2 のデータの属性情報を有する第 3 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 2 のタプルを読み出し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信し、前記記憶空間に書き込まれた前記第 4 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 4 のタプルを読み出し、検索された前記第 4 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信するステップと、

第 3 のデータと前記第 3 のタプルを指定するための属性条件を有する第 4 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 3 のタプルのうち属性条件が当該第 4 のタプルの属性条件と一致するものを検索し、検索された前記第 3 のタプルを読み出し、検索された前記第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 4 のタプルを送信するステップと、

をコンピュータに実行させるためのプログラム。

10

20

30

40

50

【請求項 6】

前記第 1 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 1 のタプルの属性情報を包含するものを検索し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 1 のタプルを送信するステップをさらにコンピュータに実行させることを特徴とする、請求項 5 に記載のプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、タプル空間を利用した間接通信を行う間接通信システム、間接通信方法、及びプログラムに関する。

【背景技術】

【0002】

ユビキタスサービスなど複数の端末を用いたネットワークで提供されるサービスの開発・運用を効率化するためには、サービスを構成するプログラム/端末の追加・変更・削除に対して柔軟に対応できる通信プラットフォーム（通信網）が必要である。この通信プラットフォームの要求条件を満足する通信モデルの候補としてTupleSpace（タプル空間）（非特許文献 1 参照）がある。TupleSpaceでは複数のプログラム間で論理的に一つの空間を共有し、各プログラム同士は、その共有空間に対してデータを読み書きすることによって間接的に通信する。

【0003】

TupleSpaceを用いた通信では、各端末あるいはサーバ上で動作するTupleSpace用のプログラム同士がTupleSpace（あるいはTupleStorage；タプルストレージ）に対してTuple（タプル；Key-Value Pair＝キーと値の組）を読み書きすることで間接的に通信する。

【0004】

具体的には送信端末の書き込み用Tuple（Actual；アクチュアル）と受信端末の読み出し用Tuple（Formal；フォーマル）とのマッチング処理により通信が発生する。書き込み用Tuple（Actual）は、データと、そのデータの属性情報を有する。読み出し用Tuple（Formal）は、読み出されるデータの属性条件を有する。そして、読み出し用Tuple（Formal）の属性条件に該当する属性情報を有する書き込み用Tuple（Actual）がTupleSpaceに存在する場合には、読み出し用Tuple（Formal）を登録した受信端末へその旨が通知される。なお、以下の説明では、ある属性条件に該当する属性情報を有するTupleを検索する処理を「マッチング処理」という。また、以下の説明では、マッチング処理において属性条件に属性情報が適合することを「マッチングする」という。

【0005】

図 1 にTupleSpaceの通信モデルの具体的な一例を示す。図 1 に示す通信モデルでは、サーバ端末 1 と、センサ端末 2 と、アクチュエータ端末 3 とが、ユビキタスネットワーク 4 に接続されていて、各端末 1～3 上で実行されているプログラム 1 1、2 1、3 1 および各端末 1～3 あるいは図示していない他の端末内のメモリなどの記憶手段と、通信手段としてのユビキタスネットワーク 4 とによって、各端末 1～3 のプログラム 1 1、2 1、3 1 などの間で論理的に共有される一つの空間（メモリ空間）であるTupleSpace 5 が構成されている。プログラム同士がこの共有空間であるTupleSpace 5 に対してデータを読み書きすることで間接的に通信が行われる。

【0006】

この図に示す例では、例えばセンサ端末 2 のプログラム 2 1 がTupleSpace 5 に登録した書き込み用Tuple（Actual；アクチュアル）6 1 とサーバ端末 1 のプログラム 1 1 がTupleSpace 5 に登録した読み出し用Tuple（Formal；フォーマル）6 2 とがマッチングした場合に、書き込み用Tuple（Actual）6 1 がサーバ端末 1 のプログラム 1 1 によって読み出される。あるいは、アクチュエータ端末 3 のプログラム 3 1 がTupleSpace 5 に登録した読み出し用Tuple（Formal）6 3 とサーバ端末 1 のプログラム 1 1 がTupleSpace 5 に登録した

書き込み用Tuple (Actual) 6 4 とがマッチングした場合に、書き込み用Tuple (Actual) 6 4 がアクチュエータ端末 3 0 のプログラム 3 1 によって読み出されることになる。なお、Tupleの凡例を参照符号 6 0 で示した。

【 0 0 0 7 】

TupleSpaceの通信モデルでは、TupleSpaceを介して任意の二つの端末のプログラム間での通信が可能であり、それらの通信はすべて独立している。例えば、センサ端末とアクチュエータ端末のプログラム同士もTupleSpaceを介して通信可能であるし、同一端末のプログラム同士もTupleSpaceを介して通信可能である。また、センサ端末とサーバ端末のプログラム間の通信とサーバ端末とアクチュエータ端末のプログラム間の通信は独立して行われる。

10

【 0 0 0 8 】

なお、TupleSpaceの通信モデルは、図 1 に示すように、センサ、アクチュエータ、サーバ等の各端末間のプログラム同士が、独立して通信を行う形態のほか、たとえば、サーバ端末を管理サーバとして用いてその管理サーバ上で実行されているプログラムを介して各端末が通信を行う形態をとることも可能である。この場合、たとえば、サーバ端末のプログラムがセンサ端末のプログラムとアクチュエータ端末のプログラムとの間の通信を仲介することで通信が行われる。設計者は、各プログラムの内容を変更することによって、どのような形態を採用するか決定することができる。

【 0 0 0 9 】

なお、TupleSpaceの通信モデルを用いた通信システムには、下記の二つの特徴がある。

20

(特徴 1) 永続的な空間 (TupleSpace) を介してデータ (Tuple) 送受が行なわれるため、通信相手のプログラムが同時に動作している必要がない。

(特徴 2) データ (Tuple) 内容のマッチングにより宛先が決定されるため、通信相手のアドレスを事前に把握している必要がない。

【 0 0 1 0 】

しかしながら、このような従来のTupleSpaceの通信モデルを用いた通信システムでは、読み出し用Tuple (Formal) を登録する端末 (受信端末) 主導の通信となっており、書き込み用Tuple (Actual) を登録する端末 (送信端末) が受信端末の存否を認識することや、送信端末が読み出し用Tuple (Formal) を読み出すことは実現されていなかった。

30

【 0 0 1 1 】

このような問題に対し、TeenyLIME (非特許文献 2 参照) は、探索用Tuple (Capability Tuple) を用いることによってこのような問題の解決を図っている。探索用Tuple (Capability Tuple) は、書き込み用Tuple (Actual) を登録する送信端末が生成し得るデータの属性情報を有する。探索用Tuple (Capability Tuple) の属性情報に読み出し用Tuple (Formal) の属性条件がマッチングした場合、探索用Tuple (Capability Tuple) を登録した送信端末によって、この読み出し用Tuple (Formal) が読み出される。このような探索用Tuple (Capability Tuple) を用いることによって、書き込み用Tuple (Actual) を登録しようとする端末 (送信端末) は、この書き込み用Tuple (Actual) のデータを読み出す可能性のある読み出し用Tuple (Formal) を読み出すことや、このような読み出し用Tuple (Formal) を登録した受信端末の存否を確認することが可能となる。

40

【 0 0 1 2 】

TeenyLIMEによれば、送信端末が受信端末の存否を確認することができるため、受信端末が現れるまでデータを生成・送信する必要が無く、データの生成・送信の処理回数を低減させることによって、バッテリーで駆動している端末の寿命 (動作可能時間) を延伸することが可能となる。また、TeenyLIMEによれば、受信端末によって登録された読み出し用Tuple (Formal) を送信端末が読み出すことが可能であるため、受信端末は送信端末へのデータの要求とデータのFormalとを、単一の読み出し用Tuple (Formal) を登録することのみによって実現することが可能となる。

【 非特許文献 1 】 D. Gelernter, "Generative communication in linda," ACM Trans.Pro

50

g.Lang.Syst., vol.7, no.1, pp.80-112, Jan.1985.

【非特許文献2】Paolo Costa, Luca Mottola, Amy L.Murphy, and Gian Pietro Picco, "Teeny LIME: Transiently Shared Tuple Space Middleware for Wireless Sensor Networks," In Proceedings of the International Workshop on Middleware for Sensor Networks, co-located with the 7th International Middleware Conference (Middleware), Melbourne, Australia, November, 2006.

【発明の開示】

【発明が解決しようとする課題】

【0013】

しかしながら、TeenyLIMEでは、受信端末が読み出し用Tuple (Formal) を登録する度に、探索用Tuple (Capability Tuple) と新たに登録された読み出し用Tuple (Formal) とのマッチング処理のみならず、読み出す必要のない書き込み用Tuple (Actual) と新たに登録された読み出し用Tuple (Formal) とのマッチング処理までもが実行されてしまう。このような、読み出す必要のない書き込み用Tuple (Actual) とのマッチング処理は無駄となってしまう。

10

【0014】

さらに、受信端末が、読み出し用Tuple (Formal) の登録によって書き込み用Tuple (Actual) の読出を意図していない場合 (例えば、センサ端末2の設定変更の場合や、アクチュエータ端末3の制御命令の送信の場合) には、上述したように登録した際に行われるマッチング処理のみならず、その後に行われるマッチング処理 (その後新たに登録された書き込み用Tuple (Actual) とのマッチング処理) も無駄となってしまう。

20

【0015】

TupleSpaceにおけるマッチング処理の効率は、検索対象となるTupleの総数、及びマッチング処理の発生頻度に依存する。また、TupleSpaceに存在するTupleの総数に占める各Tuple種別 (TeenyLIMEにおいては、Actual、Formal、Capability Tuple、の三つのTuple種別がある) の割合は、書き込み用Tuple (Actual) の割合が多数を占める。これは、読み出し用Tuple (Formal) 及び探索用Tuple (Capability Tuple) がプログラム毎に登録されるのに対し、書き込み用Tuple (Actual) はデータが生成される度に登録されるためである。このように、書き込み用Tuple (Actual) の数は大変多いため、上述したような書き込み用Tuple (Actual) との不必要なマッチング処理の発生により、TupleSpaceにおけるマッチング処理の効率が低下してしまうという問題があった。

30

【0016】

上記事情に鑑み、本発明は、書き込み用Tuple (Actual) との不必要なマッチング処理を削減しマッチング処理の効率を向上させることを可能とする間接通信システム、間接通信方法、及びプログラムを提供することを目的とする。

【課題を解決するための手段】

【0017】

[1] 上記の課題を解決するため、本発明の一態様は、複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタプルを読み書きすることで通信を行う間接通信システムであって、第1のデータと当該第1のデータの属性情報とを有する第1のタプルを管理する装置と、所望するデータの属性条件を有する第2のタプルを管理する装置と、所望する前記第2のタプルの属性条件、又は、任意の前記第2のタプルを管理する装置に読み出される第2のデータと当該第2のデータの属性情報を有する第3のタプルを管理する装置と、第3のデータと前記第3のタプルを指定するための属性条件を有する第4のタプルを管理する装置と、を備え、前記第1のタプルを管理する装置は、前記第1のタプルを前記記憶空間に書き込み、前記第2のタプルを管理する装置は、前記第2のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第1のタプルの属性情報を当該第2のタプルの属性条件で検索し、検索された前記第1のタプルを読み出し、前記記憶空間に書き込まれた前記第3のタプルのうち属性条件が当該第2のタプルの属性条件と一致するものを検索し、検索された前記第3の

40

50

タブルを読み出し、検索された前記第3のタブルを前記記憶空間に書き込んだ装置に当該第2のタブルを送信し、前記第3のタブルを管理する装置は、前記第3のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第2のタブルのうち属性条件が当該第3のタブルの属性条件と一致するものを検索し、検索された前記第2のタブルを読み出し、検索された前記第2のタブルを前記記憶空間に書き込んだ装置に当該第3のタブルを送信し、前記記憶空間に書き込まれた前記第4のタブルのうち属性条件が当該第3のタブルの属性条件と一致するものを検索し、検索された前記第4のタブルを読み出し、検索された前記第4のタブルを前記記憶空間に書き込んだ装置に当該第3のタブルを送信し、前記第4のタブルを管理する装置は、前記第4のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第3のタブルのうち属性条件が当該第4のタブルの属性条件と一致するものを検索し、検索された前記第3のタブルを読み出し、検索された前記第3のタブルを前記記憶空間に書き込んだ装置に当該第4のタブルを送信する。

10

【0018】

[2] 本発明の一態様は、上述した間接通信システムであって、前記第1のタブルを管理する装置は、前記第1のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第2のタブルのうち属性条件が当該第1のタブルの属性情報を包含するものを検索し、検索された前記第2のタブルを前記記憶空間に書き込んだ装置に当該第1のタブルを送信することを特徴とする。

【0019】

[3] 本発明の一態様は、複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタブルを読み書きすることで通信を行う間接通信システムにおいて、第1のデータと当該第1のデータの属性情報とを有する第1のタブルを管理する装置が、前記第1のタブルを前記記憶空間に書き込むステップと、所望するデータの属性条件を有する第2のタブルを管理する装置が、前記第2のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第1のタブルの属性情報を当該第2のタブルの属性条件で検索し、検索された前記第1のタブルを読み出し、前記記憶空間に書き込まれた前記第3のタブルのうち属性条件が当該第2のタブルの属性条件と一致するものを検索し、検索された前記第3のタブルを読み出し、検索された前記第3のタブルを前記記憶空間に書き込んだ装置に当該第2のタブルを送信するステップと、所望する前記第2のタブルの属性条件、又は、任意の前記第2のタブルを管理する装置に読み出される第2のデータと当該第2のデータの属性情報を有する第3のタブルを管理する装置が、前記第3のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第2のタブルのうち属性条件が当該第3のタブルの属性条件と一致するものを検索し、検索された前記第2のタブルを読み出し、検索された前記第2のタブルを前記記憶空間に書き込んだ装置に当該第3のタブルを送信し、前記記憶空間に書き込まれた前記第4のタブルのうち属性条件が当該第3のタブルの属性条件と一致するものを検索し、検索された前記第4のタブルを読み出し、検索された前記第4のタブルを前記記憶空間に書き込んだ装置に当該第3のタブルを送信するステップと、第3のデータと前記第3のタブルを指定するための属性条件を有する第4のタブルを管理する装置が、前記第4のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第3のタブルのうち属性条件が当該第4のタブルの属性条件と一致するものを検索し、検索された前記第3のタブルを読み出し、検索された前記第3のタブルを前記記憶空間に書き込んだ装置に当該第4のタブルを送信するステップと、を含む間接通信方法である。

20

30

40

【0020】

[4] 本発明の一態様は、上述した間接通信方法であって、前記第1のタブルを管理する装置が、前記第1のタブルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第2のタブルのうち属性条件が当該第1のタブルの属性情報を包含するものを検索し、検索された前記第2のタブルを前記記憶空間に書き込んだ装置に当該第1のタブルを送信するステップをさらに含むことを特徴とする。

【0021】

50

〔 5 〕本発明の一態様は、複数の端末が共通の通信網に接続され、前記通信網上に存在する記憶空間に対して属性名と属性値のペアであるタプルを読み書きすることで通信を行う間接通信システムにおいて、第 1 のデータと当該第 1 のデータの属性情報とを有する第 1 のタプルを前記記憶空間に書き込むステップと、所望するデータの属性条件を有する第 2 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 1 のタプルの属性情報を当該第 2 のタプルの属性条件で検索し、検索された前記第 1 のタプルを読み出し、前記記憶空間に書き込まれた前記第 3 のタプルのうち属性条件が当該第 2 のタプルの属性条件と一致するものを検索し、検索された前記第 3 のタプルを読み出し、検索された前記第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 2 のタプルを送信するステップと、所望する前記第 2 のタプルの属性条件、又は、任意の前記第 2 のタプルを管理する装置に読み出される第 2 のデータと当該第 2 のデータの属性情報を有する第 3 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 2 のタプルを読み出し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信し、前記記憶空間に書き込まれた前記第 4 のタプルのうち属性条件が当該第 3 のタプルの属性条件と一致するものを検索し、検索された前記第 4 のタプルを読み出し、検索された前記第 4 のタプルを前記記憶空間に書き込んだ装置に当該第 3 のタプルを送信するステップと、第 3 のデータと前記第 3 のタプルを指定するための属性条件を有する第 4 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 3 のタプルのうち属性条件が当該第 4 のタプルの属性条件と一致するものを検索し、検索された前記第 3 のタプルを読み出し、検索された前記第 3 のタプルを前記記憶空間に書き込んだ装置に当該第 4 のタプルを送信するステップと、をコンピュータに実行させるためのプログラムである。

【 0 0 2 2 】

〔 6 〕本発明の一態様は、上述したプログラムであって、前記第 1 のタプルを前記記憶空間に書き込む場合に、前記記憶空間に書き込まれた前記第 2 のタプルのうち属性条件が当該第 1 のタプルの属性情報を包含するものを検索し、検索された前記第 2 のタプルを前記記憶空間に書き込んだ装置に当該第 1 のタプルを送信するステップをさらにコンピュータに実行させることを特徴とする。

【発明の効果】

【 0 0 2 3 】

本発明により、書き込み用 Tuple (Actual) との不必要なマッチング処理を削減し、マッチング処理の効率を向上させることが可能となる。

【発明を実施するための最良の形態】

【 0 0 2 4 】

本発明における TupleSpace は、図 1 を参照して説明したタプル空間 TupleSpace の通信モデルを独自に拡張したものであり、タプル種別としては actual 及び formal に加えて、capable 及び reachable が用いられる。

【 0 0 2 5 】

actual (第 1 のタプル) は、送信端末が TupleSpace にデータを書き込む際に用いられ、データと当該データの属性情報 (データ属性情報) とを有する。formal (第 2 のタプル) は、受信端末が TupleSpace から actual を読み出すために用いられ、読み出される actual が有するデータの属性条件を有する。capable (第 3 のタプル) は、TupleSpace から formal を読み出すために用いられ、読み出される formal が有する属性条件が、任意の formal によって読み出されうるデータと当該データのデータ属性情報を有する。reachable (第 4 のタプル) は、TupleSpace から capable を読み出すために用いられ、データと当該データを読み出す capable を指定するための属性条件とを有する。

【 0 0 2 6 】

図 2 に本発明の実施の形態による TupleSpace の通信モデルを示す。上述したように、各端末上で動作するプログラム同士 (1 1 0 , 2 1 0 , 3 1 0) は TupleSpace (あるいは Tu

pleStorage) 50 に対して Tuple (Key-Value Pair) を読み書きすることで間接的に通信する。なお、図 2 は本実施の形態の通信モデルを模式的に示したものであり、この通信モデルを実現する通信システムには各端末 10 ~ 30 がそれぞれ複数台接続されても良い。

【0027】

図 2 に示す通信モデルでは、サーバ端末 10 と、センサ端末 20 と、アクチュエータ端末 30 とが、ユビキタスネットワーク 40 に接続されている。各端末 10 ~ 30 上で実行されているプログラム 110、210、310 および各端末 10 ~ 30 あるいは図示していない他の端末内のメモリなどの記憶手段と、通信手段としてのユビキタスネットワーク 40 とによって、各端末 10 ~ 30 のプログラム 110、210、310 などの間で論理的に共有される一つの空間 (メモリ空間) である TupleSpace 50 が構成されている。プログラム同士がこの共有空間である TupleSpace 50 に対してデータを読み書きすることで、各端末間の通信が間接的に行われる。このような TupleSpace 50 は、ユビキタスネットワーク 40 に接続された各端末の記憶領域のうち他の端末にアクセス可能 (読み書き可能) に解放された記憶領域や、ユビキタスネットワーク 40 に接続され各端末 10 ~ 30 によってアクセス可能 (読み書き可能) に構成された記憶装置の記憶領域によって構成される。

【0028】

なお、TupleSpace の通信モデルでは、TupleSpace 50 を介して任意の二つの端末のプログラム間での通信が可能であり、それらの通信はすべて独立している。例えば、センサ端末 20 とアクチュエータ端末 30 のプログラム同士 210 と 310 は、TupleSpace 50 を介して通信可能である。また、同一端末の異なるプログラム同士も TupleSpace 50 を介して通信可能である。また、センサ端末 20 とサーバ端末 10 とのプログラム間の通信と、サーバ端末 10 とアクチュエータ端末 30 とのプログラム間の通信は独立して行われる。

【0029】

また、TupleSpace 50 の通信モデルは、図 2 に示すように、サーバ端末 10、センサ端末 20、アクチュエータ端末 30 等の各端末間のプログラム同士が独立して通信を行う形態のほか、他の通信形態によって実現されることも可能である。例えば、サーバ端末 10 が管理サーバとして動作することによってその管理サーバ上で実行されているプログラムを介して他の各端末が通信を行う形態をとることも可能である。この場合、サーバ端末 10 のプログラム 110 がセンサ端末 20 のプログラム 210 とアクチュエータ端末 30 のプログラム 310 との間の通信を仲介することによって、センサ端末 20 とアクチュエータ端末 30 との間の通信が実現される。TupleSpace 50 においてどのような形態が採用されるかは、設計者が適宜選択することができる。

【0030】

図 3 は、各タプル種別の書式の具体例を表す図である。actual は、「date」、「position」、「address」、「subject」、「type」、「body」の項目を有する。「date」は、タイムスタンプ情報を表す項目であり、時刻情報を示す値が対応付けられる。「position」は、位置情報を表す項目であり、この actual を TupleSpace 50 に登録した端末 (以下、「送信端末」という) の位置に関する情報を示す値が対応付けられる。「address」は、アドレス情報を表す項目であり、送信端末のアドレスを示す値が対応付けられる。ここで、アドレス情報とは、例えばネットワークにおける論理的なアドレス情報であっても良く、例えば IP アドレスや MAC アドレスやネットワーク名などであっても良い。また、アドレス情報とは、例えば端末が設置された物理的な場所を表す情報であっても良く、例えば住所や建物名などであっても良い。「subject」は、端末の種別情報を表す項目であり、送信端末の端末種別を示す値が対応付けられる。「type」は、端末へのコマンドやイベントの種別を表すキーであり、それらを表す値が対応づけられている。「body」は、センサ出力などの「パラメータ」 (あるいはデータ) を表すキーであり、プログラム間で規定された任意のフィールドの集合が対応づけられている。この「body」に設定される値が送信されるデータとなる。

【0031】

10

20

30

40

50

formal、capable、reachableは、「date_range」、「position_range」、「address_range」、「subject(N-gram)」、「type(N-gram)」、「body」の項目を有する。「date_range」は、読み出し条件（読み出されるデータの属性条件）のタイムスタンプ情報を表す項目であり、時刻の範囲を示す情報（範囲）を示す値が対応付けられる。「position_range」は、読み出し条件の位置情報を表す項目であり、読み出そうとするデータを有するactualをTupleSpace 50に登録した端末（即ち送信端末）の位置の範囲に関する情報（範囲）を示す値が対応付けられる。「address_range」は、読み出し条件のアドレス情報を表す項目であり、送信端末のアドレスの範囲を示す値が対応付けられる。「subject(N-gram)」は、読み出し条件の端末の種別情報を表す項目であり、送信端末の端末種別を示す値（部分一致を含む）が対応付けられる。「type(N-gram)」は、読み出し条件のコマンドやイベントの種別を表すキーであり、それら（種別）を表す値が対応づけられている。また、formal、capable、reachableは、それぞれ自身をTupleSpace 50へ登録した端末（以下、「登録端末」という）に対してデータを送信するためのアドレスの情報も有する。他の端末は、formal、capable、reachableが有するこのアドレスの情報を参照することによって、このTupleをTupleSpace 50に登録した端末に対してデータを送信すること（例えば、Tupleを通知することなど）が可能となる。また、formal、capable、reachableは、それぞれ自身がTupleSpace 50へ登録された時刻に関する情報も有する。他の端末は、formal、capable、reachableそれぞれについて、いずれが最新のものであるかを判断する際に、この登録された時刻に関する情報を参照することによって判断を行う。

10

20

30

40

50

【0032】

図4は、本発明のTupleSpaceにおける各タプル種別の関係を表す図である。図4において、publisher 60、subscriber 70、listener 80、caster 90は、それぞれ通信プラットフォームに対応するタプル種別を登録する主体、即ち図2におけるプログラムを表す。従って、図2におけるサーバ端末10、センサ端末20、アクチュエータ端末30は、それぞれpublisher 60、subscriber 70、listener 80、caster 90いずれか一つ又は複数のプログラムを備えることにより、TupleSpace 50を介して間接的に通信を行う。なお、publisher 60が搭載された端末が本発明における第1のタプルを管理する装置に相当し、subscriber 70が搭載された端末が本発明における第2のタプルを管理する装置に相当し、listener 80が搭載された端末が本発明における第3のタプルを管理する装置に相当し、caster 90が搭載された端末が本発明における第4のタプルを管理する装置に相当する。

【0033】

publisher 60は、actualをTupleSpace 50に登録するプログラムである。subscriber 70は、formalをTupleSpace 50に登録するプログラムである。listener 80は、capableをTupleSpace 50に登録するプログラムである。caster 90は、reachableをTupleSpace 50に登録するプログラムである。なお、複数のプログラムがまとまった一つのプログラムとして構成されても良い。

【0034】

図4において、Data、Formality、Capable、Reachabilityは、それぞれプログラムが取り扱うデータ種別を表す。各データ種別の内容は、プログラムによってTupleSpace 50に登録される際にそれぞれ対応するタプル種別に書式が変更される。

【0035】

図4において、writeは、TupleSpace 50へのTupleの登録（書き込み）処理を表す。readは、TupleSpace 50からの能動的なTupleの読み出し処理を表す。takeは、TupleSpace 50からの能動的なTupleの読み出し処理の亜種である移動処理を表す。soundは、TupleSpace 50からの受動的なTupleの読み出し処理を表す。移動処理とは、TupleSpace 50からTupleを読み出すのと同時に、TupleSpace 50から当該Tupleを消去する処理である。各プログラム60～90は、TupleSpace 50からの通知の種別に基づいて、能動的処理（read及びtake）と、受動的処理（sound）とを区別して行う。また、readとtakeのいずれを行うべきかについては、formalを登録する際にsubscriber 70が決定する。

【 0 0 3 6 】

図 4 において、matchingの矢印は、その矢印によって連結された二つのTupleの間でマッチング処理が行われることを表す。ここで、マッチング処理とは、属性条件に属性情報が適合するか否か、又は属性条件と属性条件とが適合するか否か、を判定する処理を表す。また、属性条件に属性情報が適合すること、又は属性条件と属性条件とが適合することをマッチングするという。矢印の向きは、マッチング処理の方向を表し、矢印の先のTupleにおける属性情報又は属性条件が、矢印の元のTupleにおける属性条件に包含されるか否か（適合するか否か）がマッチング処理によって判断される。

【 0 0 3 7 】

writeは、その主体に応じて、publish、subscribe、listen、castに分類される。publisherは、publisher 6 0 がTupleSpace 5 0 にDataをactualとしてwriteする処理を表す。subscriberは、subscriber 7 0 がTupleSpace 5 0 にFormalityをformalとしてwriteする処理を表す。listenは、listener 8 0 がTupleSpace 5 0 にCapabilityをcapableとしてwriteする処理を表す。castは、caster 9 0 がTupleSpace 5 0 にReachabilityをreachableとしてwriteする処理を表す。

【 0 0 3 8 】

次に、各プログラム 6 0 ~ 9 0 の動作について説明する。

publisher 6 0 は、publishの際に、actualの内容とその生存期間（limit）を設定する。生存期間は、そのactualの有効期間を表わす。生存期間が満了するまでの間、actualはformalのマッチング処理の対象となる。また、publisher 6 0 は、publishの際に、actualの属性情報に基づいてformalを逆検索する。

【 0 0 3 9 】

図 5 は、publisher 6 0 の動作例を表すフローチャートである。まず、publisher 6 0 は、登録しようとするactualをread又はtakeするformalを逆検索する（ステップ S 5 0 1）。具体的には、publisher 6 0 は、自身が登録しようとするactualの属性情報に適合する属性条件を有するformalを検索する。このようなformalが存在しない場合（ステップ S 5 0 2 - N O）、publisher 6 0 は、actualをTupleSpace 5 0 に登録し（ステップ S 5 0 3）、処理を終了する。

【 0 0 4 0 】

一方、このようなformalが存在する場合（ステップ S 5 0 2 - Y E S）、publisher 6 0 は、このformalをTupleSpace 5 0 に登録したsubscriber 7 0 に対し、自身が登録しようとしているTupleの通知を行う（ステップ S 5 0 4）。通知とは、Tupleの内容を送信することであり、Tupleがactualである場合は、actualのデータ、このデータの属性情報、actualの各設定内容（生存期間など）を送信することである。

【 0 0 4 1 】

次に、publisher 6 0 は、このformalの設定が、readであるかtakeであるかについて判定する。readである場合（ステップ S 5 0 5 - N O）、publisher 6 0 は、actualをTupleSpace 5 0 に登録する（ステップ S 5 0 6）。一方、takeである場合（ステップ S 5 0 5 - Y E S）、publisher 6 0 はactualをTupleSpace 5 0 に登録せずに処理を終了する。なお、takeについては後述する。

【 0 0 4 2 】

subscriber 7 0 は、subscribeの際に、formalの内容とその生存期間との設定、actual / capableのread / takeの設定、one / allの設定を行う。one / allの設定とは、通知されるTupleを、マッチングしたTupleの全てとするか（all）、マッチングしたTupleの内の一つ（本実施形態の場合は、最後にTupleSpace 5 0 に登録されたTuple（即ち最新のTuple））とするか（one）を設定することである。

【 0 0 4 3 】

図 6 は、subscriber 7 0 の動作例を表すフローチャートである。まず、subscriber 7 0 は、登録しようとするformalをreadする最新のcapableを逆検索する（ステップ S 6 0 1）。具体的には、subscriber 7 0 は、自身が登録しようとするformalの属性条件に適合す

10

20

30

40

50

る属性条件を有するcapableのうち最新のものを検索する。このようなcapableが存在する場合（ステップS 6 0 2 - Y E S）、subscriber 7 0はこのcapableを登録したlistener 8 0に対し、自身が登録しようとしているformalについて通知する（ステップS 6 0 3）。

【 0 0 4 4 】

このようなcapableが存在しない場合（ステップS 6 0 2 - N O）、又はステップS 6 0 3の処理の後、subscriber 7 0は、自身が登録しようとしているformalの設定がallであるか否か判断する。設定がallである場合（ステップS 6 0 4 - Y E S）、subscriber 7 0は、登録しようとするformalの属性条件にマッチングする全てのactual及びcapableを検索する（ステップS 6 0 5）。

10

【 0 0 4 5 】

次に、subscriber 7 0は、マッチングしたcapableのうち、quietの設定がされているものを除外し（ステップS 6 0 6）、残った全てのactual及びcapableをTupleSpace 5 0から読み出す（ステップS 6 0 7）。そして、subscriber 7 0は、読み出したactual及びcapableそれぞれについてtake設定がなされているか否かを判断し、takeの設定がされているactual及びcapableをTupleSpace 5 0から削除する（ステップS 6 0 8 - Y E S、ステップS 6 0 9）。このとき、subscriber 7 0は、takeの設定がされていないactual及びcapableについては削除を行わない（ステップS 6 0 8 - N O）。そして、subscriber 7 0は、formalをTupleSpace 5 0に登録して処理を終了する（ステップS 6 1 0）。

【 0 0 4 6 】

ステップS 6 0 4の処理において、設定がallでない場合（ステップS 6 0 4 - N O）、subscriber 7 0は、自身が登録しようとしているformalとマッチングするactualか、又はnoise設定のcapableのうち最後に登録されたものを検索する（ステップS 6 1 1）。このようなTupleが存在しなかった場合（ステップS 6 1 2 - N O）、subscriber 7 0はformalをTupleSpace 5 0に登録して処理を終了する（ステップS 6 1 0）。

20

【 0 0 4 7 】

ステップS 6 1 2においてTupleが存在した場合（ステップS 6 1 2 - Y E S）、subscriber 7 0は、この検索されたTupleを読み出す（ステップS 6 1 3）。そして、subscriber 7 0は、読み出したactual及びcapableそれぞれについてtake設定か否かを判断し、takeの設定がされているactual及びcapableをTupleSpace 5 0から削除し処理を終了する（ステップS 6 1 4 - Y E S、ステップS 6 1 5）。このとき、subscriber 7 0は、takeの設定がされていないactual及びcapableについては削除を行わずに処理を終了する（ステップS 6 1 4 - N O）。

30

【 0 0 4 8 】

listener 8 0は、listenの際に、capableの内容とその生存期間との設定、one / allの設定、capableの受動的な読み出しを許可するか否か（noise / quiet）の設定を行う。capableがquietに設定されると、subscriber 7 0とcaster 9 0がTupleSpace 5 0におけるcapableを検索・読み出しする際に、このcapableは検索されたとしても読み出されない。

【 0 0 4 9 】

図 7 は、listener 8 0の動作例を表すフローチャートである。まず、listener 8 0は、自身が登録しようとするcapableの設定がnoiseであるか否か判断する（ステップS 7 0 1）。設定がnoiseである場合（ステップS 7 0 1 - Y E S）、listener 8 0は、登録しようとするcapableをread又はtakeする最新のformalを逆検索する（ステップS 7 0 2）。具体的には、listener 8 0は、自身が登録しようとするcapableの属性条件に適合する属性条件を有する最新のformalを検索する。このようなformalが見つかった場合（ステップS 7 0 3 - Y E S）、listener 8 0は、このformalをTupleSpace 5 0に登録したsubscriber 7 0に対し、登録しようとするcapableを通知する（ステップS 7 0 4）。そして、listener 8 0は、登録しようとするcapableがtake設定である場合には（ステップS 7 0 5）、このcapableをTupleSpace 5 0に登録することなく処理を終える。

40

【 0 0 5 0 】

50

ステップ S 7 0 1 においてnoise設定でない場合 (ステップ S 7 0 1 - N O)、又はステップ S 7 0 3 において逆検索の対象となる formal が存在しない場合 (ステップ S 7 0 3 - N O)、又はステップ S 7 0 5 においてtake設定でない場合 (ステップ S 7 0 5 - N O)、listener 8 0 は、登録しようとする capable が all 設定であるか否か判断する (ステップ S 7 0 6)。all 設定である場合 (ステップ S 7 0 6 - Y E S)、listener 8 0 は、登録しようとする capable とマッチングする全ての formal 及び reachable を検索する (ステップ S 7 0 7)。listener 8 0 は、マッチングした全ての formal 及び reachable を読み出す (ステップ S 7 0 8)。そして、listener 8 0 は、capable を TupleSpace 5 0 に登録し (ステップ S 7 0 9)、処理を終了する。

【 0 0 5 1 】

10

ステップ S 7 0 6 において、登録しようとする capable が all 設定でない場合 (ステップ S 7 0 6 - N O)、listener 8 0 は、登録しようとする capable とマッチングする formal 及び reachable のうち、最後に登録されたものを検索する (ステップ S 7 1 0)。このような Tuple が存在しない場合 (ステップ S 7 1 1 - N O)、listener 8 0 は、capable を TupleSpace 5 0 に登録し (ステップ S 7 0 9)、処理を終了する。一方、このような Tuple が存在する場合 (ステップ S 7 1 1 - Y E S)、listener 8 0 は、この Tuple を読み出し (ステップ S 7 1 2)、capable を TupleSpace 5 0 に登録することなく処理を終了する。

【 0 0 5 2 】

caster 9 0 は、cast の際に、reachable の内容とその生存期間と通知される capable 数の設定を行う。通知される capable 数は、データを受信する listener 8 0 の数を表し、reachable の属性条件に基づく anycast / multicast を行うことができる。

20

【 0 0 5 3 】

図 8 は、caster 9 0 の動作例を表すフローチャートである。まず、caster 9 0 は、自身が登録しようとする reachable を read する最新の capable を逆検索する (ステップ S 8 0 1)。具体的には、caster 9 0 は、自身が登録しようとする reachable の属性条件に適合する属性条件を有する capable を検索する。このような capable が存在する場合 (ステップ S 8 0 2 - Y E S)、caster 9 0 は、この capable を登録した listener 8 0 に対し、自身が登録しようとする reachable を通知する (ステップ S 8 0 3)。

【 0 0 5 4 】

このような capable が存在しない場合 (ステップ S 8 0 2 - N O)、又はステップ S 8 0 3 の処理の後、caster 9 0 は、自身が登録しようとしている reachable の設定が all であるか否か判断する。設定が all である場合 (ステップ S 8 0 4 - Y E S)、caster 9 0 は、登録しようとする reachable の属性条件にマッチングする全ての capable を検索する (ステップ S 8 0 5)。

30

【 0 0 5 5 】

次に、caster 9 0 は、マッチングした capable のうち、quiet の設定がされているものを除外し (ステップ S 8 0 6)、残った全ての capable を TupleSpace 5 0 から読み出す (ステップ S 8 0 7)。そして、caster 9 0 は、読み出した capable それぞれについて take 設定か否かを判断し、take の設定がされている capable を TupleSpace 5 0 から削除する (ステップ S 8 0 8 - Y E S、ステップ S 8 0 9)。このとき、caster 9 0 は、take の設定がされていない capable については削除を行わない (ステップ S 8 0 8 - N O)。そして、caster 9 0 は、reachable を TupleSpace 5 0 に登録して処理を終了する (ステップ S 8 1 0)。

40

【 0 0 5 6 】

ステップ S 8 0 4 の処理において、設定が all でない場合 (ステップ S 8 0 4 - N O)、caster 9 0 は、自身が登録しようとしている reachable とマッチングし且つ noise 設定の capable のうち最後に登録されたものを検索する (ステップ S 8 1 1)。このような Tuple が存在しなかった場合 (ステップ S 8 1 2 - N O)、caster 9 0 は reachable を TupleSpace 5 0 に登録して処理を終了する (ステップ S 8 1 0)。

【 0 0 5 7 】

50

ステップ S 8 1 2 において Tuple が存在した場合（ステップ S 8 1 2 - Y E S）、caster 9 0 は、この検索された Tuple を読み出す（ステップ S 8 1 3）。そして、caster 9 0 は、読み出した capable それぞれについて take 設定か否かを判断し、take の設定がされている capable を TupleSpace 5 0 から削除し処理を終了する（ステップ S 8 1 4 - Y E S、ステップ S 8 1 5）。このとき、caster 9 0 は、take の設定がされていない capable については削除を行わずに処理を終了する（ステップ S 8 1 4 - N O）。

【 0 0 5 8 】

従来の TeenyLIME では、本発明における actual、formal、capable に類似する Tuple である、actual、formal、ct が用いられていたが、本発明における reachable に類似する Tuple は用いられていなかった。そのため、送信端末主導の通信を効率的に実現できていなかった。即ち、本発明によれば、GET 処理を formal を登録することのみによって実現することが可能となり、それに伴い TupleSpace 5 0 に登録される Tuple の総数を削減させ、マッチング処理の発生頻度を抑えることが可能となった。また、reachable のマッチング処理の対象から actual を除くことにより、即ち reachable のマッチング処理の対象から TupleSpace 5 0 に占める割合が支配的な Tuple を除くことにより、送信端末主導の通信を効率的に実現することが可能となった。さらに、reachable と formal とを同じデータ種別（Formality）として扱うことによって、プログラムを配備する端末の実装規模を縮小させることが可能となった。

【 0 0 5 9 】

また、本発明により、TeenyLIME におけるメリットをそのまま享受することが可能であり、さらには、送信端末主導のデータ通信と受信端末主導のデータ通信とをそれぞれ独立に実現させることが可能となる。また、サーバ端末からセンサ / アクチュエータ端末へデータを送信する際に、当該データをどのセンサ / アクチュエータ端末が受信すべきか、各センサ / アクチュエータ端末がどのデータを受信すべきかの指定をサーバ端末で行うことが可能となる。また、受信端末の存否確認、送信端末へのデータ要求、通常のデータの読み出しにおけるマッチング処理をそれぞれ独立させることによって検索効率の向上を図ることが可能となる。さらに、送信端末へ受信端末の Tuple を通知することによって、データがどの受信端末に読み出されたかを判断することが可能となる。

【 0 0 6 0 】

< 変形例 >

図 4 においては、listener 8 0 が reachable と formal とを同じデータ種別（Formality）として read しているが、listener 8 0 は reachable を Reachability として読み出しでも良い。また、図 4 においては、listener 9 0 が capable と formal との二つの Tuple 種別だけを処理すれば良く、caster 9 0 としての役割を必要としないプログラムを配備する端末の実装規模を縮小することが可能となる。

【 0 0 6 1 】

また、図 6 のステップ S 6 0 1 の処理において、subscriber 7 0 は、noise 設定された capable のみを検索の対象とし、このような capable を登録した listener 8 0 に対してステップ S 6 0 3 の処理において formal を通知しても良い。

【 0 0 6 2 】

また、subscriber 7 0 は、subscribe の際に、formal の受動的な読み出しを許可するかどうか（noise / quiet）の設定を行っても良い。formal が quiet に設定されると、listener 8 0 が図 7 のステップ S 7 0 7 の処理において TupleSpace 5 0 における formal を検索しステップ S 7 0 8 の処理において読み出しする際に、この formal は検索されたとしても listener 8 0 によって読み出されない。即ち、この場合、listener 8 0 は、図 7 のステップ S 7 0 8 の処理において、noise 設定された formal を読み出し、quiet 設定された formal は読み出さない。

【 0 0 6 3 】

また、図 7 のステップ S 7 0 8 の処理において、listener 8 0 は、noise 設定された capable のみを検索の対象とし、このような capable を登録した listener 8 0 に対してステッ

ブ S 6 0 3 の処理において formal を通知しても良い。

【 0 0 6 4 】

上述した実施形態における各端末の機能をコンピュータで実現するようにしても良い。その場合、これらの各機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータシステムに読み込ませ、実行することによって実現しても良い。なお、ここでいう「コンピュータシステム」とは、OS や周辺機器等のハードウェアを含むものとする。また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ROM、CD-ROM 等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムを送信する場合の通信線のように、短時間の間、動的にプログラムを保持するもの、その場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリのように、一定時間プログラムを保持しているものも含んでも良い。また上記プログラムは、前述した機能の一部を実現するためのものであっても良く、さらに前述した機能をコンピュータシステムにすでに記録されているプログラムとの組み合わせで実現できるものであっても良い。

10

以上、この発明の実施形態について図面を参照して詳述してきたが、具体的な構成はこの実施形態に限られるものではなく、この発明の要旨を逸脱しない範囲の設計等も含まれる。

【図面の簡単な説明】

20

【 0 0 6 5 】

【図 1】 TupleSpace の通信モデルの具体的な一例を表す図である。

【図 2】 本発明の実施の形態による TupleSpace の通信モデルを表す図である。

【図 3】 各タプル種別の書式の具体例を表す図である。

【図 4】 本発明の TupleSpace における各タプル種別の関係を表す図である。

【図 5】 publisher の動作例を表すフローチャートである。

【図 6】 subscriber の動作例を表すフローチャートである。

【図 7】 listener の動作例を表すフローチャートである。

【図 8】 caster の動作例を表すフローチャートである。

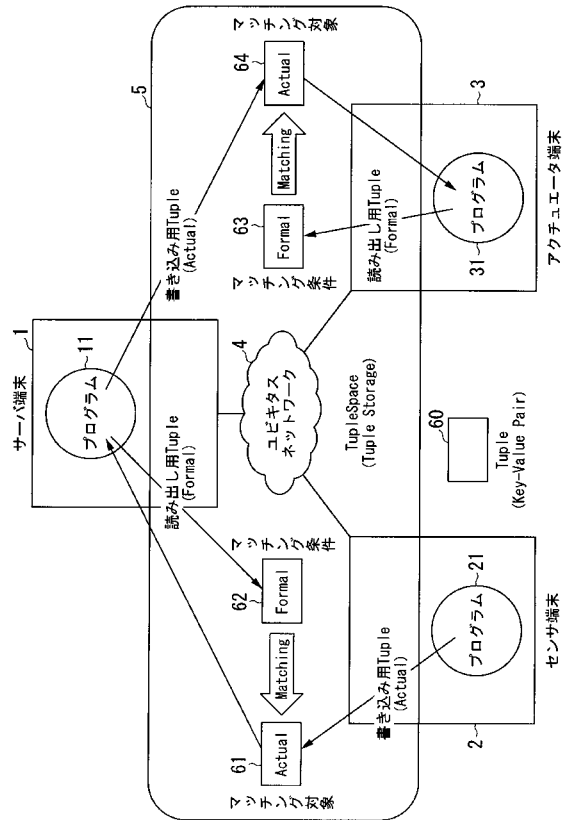
【符号の説明】

30

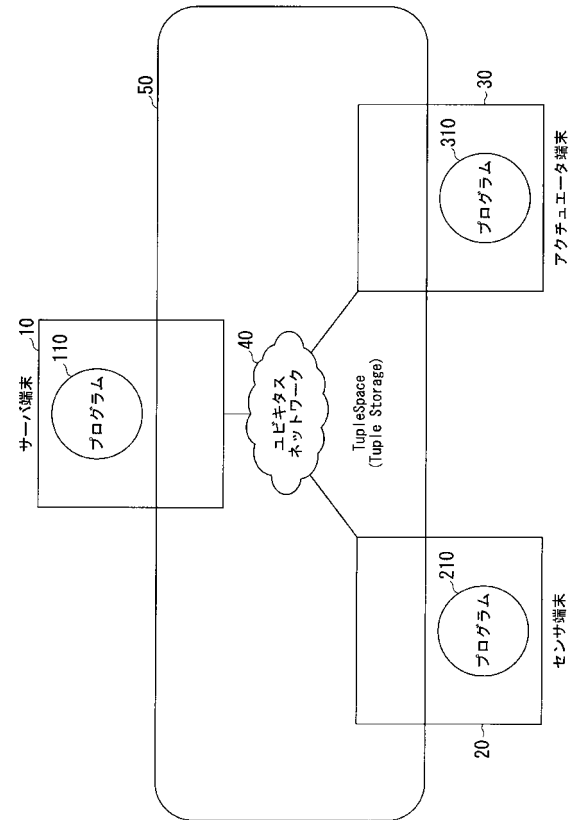
【 0 0 6 6 】

1 0 ... サーバ端末 , 2 0 ... センサ端末 , 3 0 ... アクチュエータ端末 , 1 1 0 , 2 1 0 , 3 1 0 ... プログラム , 4 0 ... ユビキタスネットワーク , 5 0 ... TupleSpace , 6 0 ... publisher , 7 0 ... subscriber , 8 0 ... listener , 9 0 ... caster

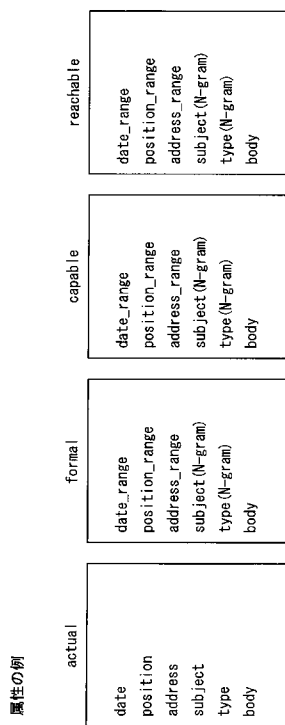
【図 1】



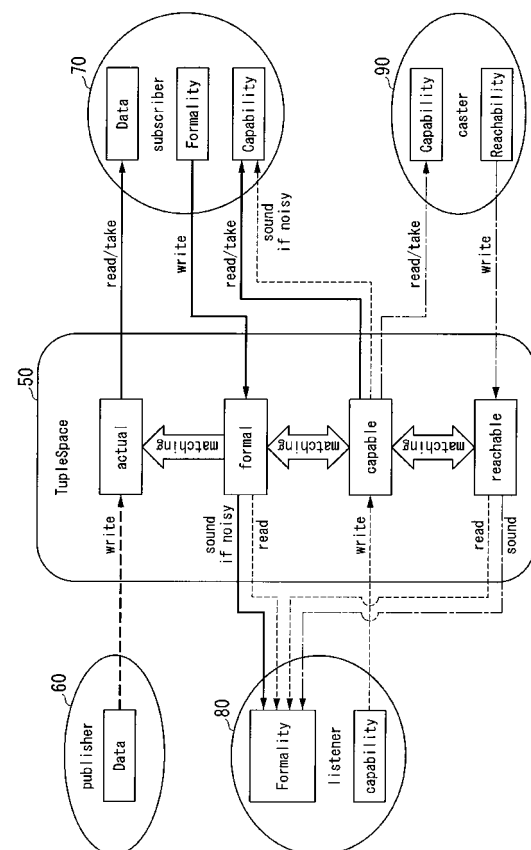
【図 2】



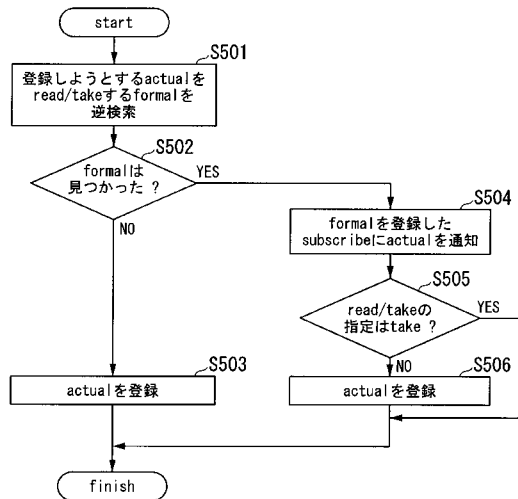
【図 3】



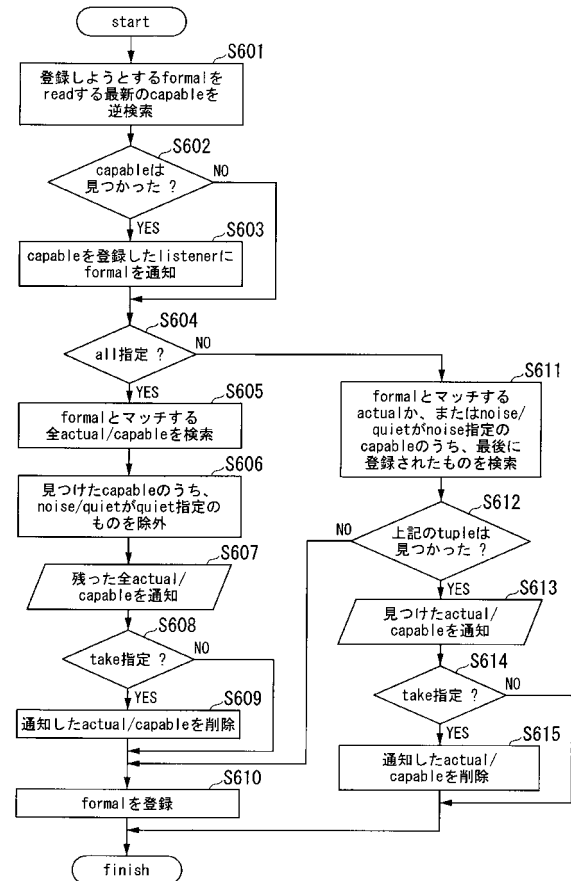
【図 4】



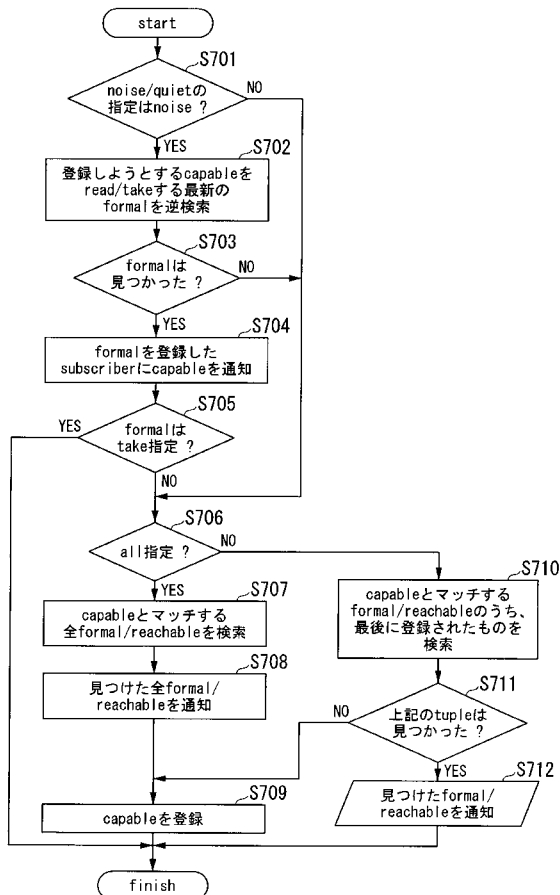
【図 5】



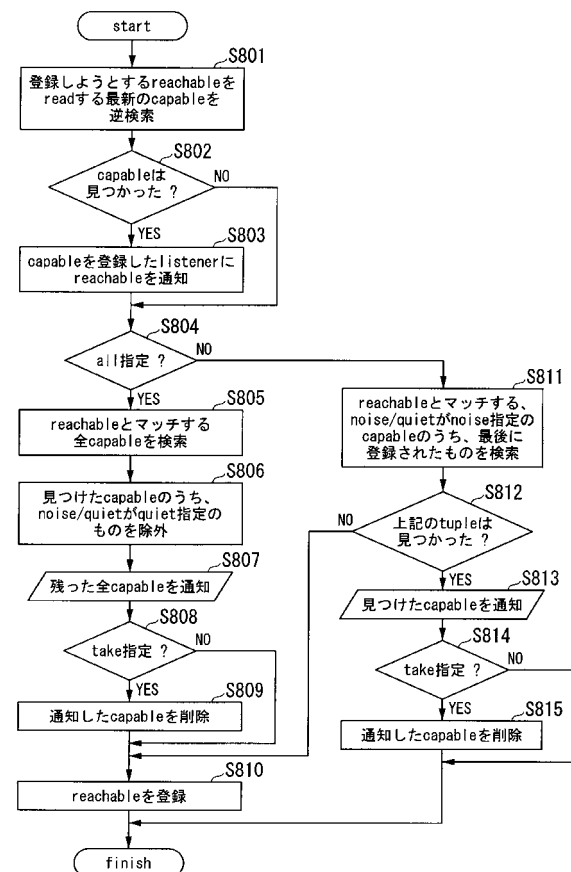
【図 6】



【図 7】



【図 8】



フロントページの続き

(72)発明者 山本 淳

東京都千代田区大手町二丁目 3 番 1 号 日本電信電話株式会社内

F ターム(参考) 5B045 BB19 BB43 BB47