

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第7部門第3区分

【発行日】平成19年12月6日(2007.12.6)

【公表番号】特表2000-514975(P2000-514975A)

【公表日】平成12年11月7日(2000.11.7)

【年通号数】公開・登録公報999999

【出願番号】特願平10-506535

【国際特許分類】

H 0 4 M 3/42 (2006.01)

H 0 4 Q 3/545 (2006.01)

【F I】

H 0 4 M 3/42 A

H 0 4 Q 3/545

【手続補正書】

【提出日】平成19年7月6日(2007.7.6)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】補正の内容のとおり

【補正方法】変更

【補正の内容】

手 続 補 正 書

平成19年 7月 6日

特 許 庁 長 官 殿



1. 事件の表示

平成10年 特許願 第506535号

2. 補正をする者

氏名又は名称 テレフオンアクチーボラゲット エル エム エリクソン
(パブル)

3. 代 理 人

〒102-0094

東京都千代田区紀尾井町3番6号

秀和紀尾井町パークビル 7F

TEL 03(5276)3241 (代表)

FAX 03(5276)3242 (代表)

(7642) 弁理士 大塚 康徳

連絡先 担当 高柳 司郎



4. 補正対象書類名

明細書及び請求の範囲

5. 補正対象項目名

全文

6. 補正の内容

別紙のとおり



方 式 審 査 (渡 辺)

明 細 書

手順の確認

- 5 この発明は、ネットワークプロトコルにおける手順を確認
 (validation) する方法に関する。特に、この発明は、欧州電気通信
 標準 ETS 300374 に定義されるケイバビリティセット 1 (CS1) の支援
 に必要なインテリジェントネットワークアプリケーションプロトコル (INAP)
 P) の様なネットワークプロトコルに関する。
- 10 INAP CS1 の様なプロトコルにおいては、特定のタスクを遂行するため
 実施される幾つかの正しい手順又は動作のシーケンスが存在する。しかし、可能
 な手順の数が大きいので、その正しい手順は直接には定義されずに、むしろ一組
 の規則により定義される。これらの規則に違反しないいかなる手順又は動作のシ
 ーケンスも、そこで正しい手順と考えられる。
- 15 先行技術においては、これらの規則は、各エンティティ、及び各インタフェー
 スに関して有限状態マシン (finite state machine) (FSM) を定義することにより記述することが出来、インタフェースは 2 つのエン
 ティティの間の境界である。この FSM は、そこで一つのプロセスの行動に対す
 るモデルとして作用する。一つの FSM は、複数の状態から成り、これらは相互
- 20 に接続出来る。一つのプロセスは、いかなる 1 つの時においても 1 つの状態にあ
 ることだけが出来るが、一つの事象の結果として 1 つの状態から接続された一つ
 の状態へ移動出来る。1 つの状態からもう 1 つの状態への移行に当たり、幾つか
 のアクションが行われる、この様なシステムは、KAKUDA 等の「特徴相互作用
 の動的解決方法及びその発展」、電気通信システムにおける特徴相互作用 II
- 25 I、第 3 特徴相互作用ワークショップにおいて提供された頁 (FIW'9,
 KYOTO, JP, Oct. 11-13, 1995) に記載されている。
- ネットワークプロトコルにおいて、手順を確認するため一つの FSM を使用す
 る時、その FSM を制御する事象がそれならそのプロトコルにおいて定義された
 動作であり、又は呼出し (call) プロセスの様な他のプロセスから来る他の

事象がこれである。

- INAP CS1の様なネットワークプロトコルにおいて正しい手順を定義するこれらの規則は、単一アソシエーション制御機能(SACF)規則及び多数アソシエーション制御機能(MACF)規則である。SACF規則は、単一のアソシエーション(結合)がある場合に適用し、MACF規則は数個の関係するアソシエーションがある場合に適用し、一つのアソシエーションは一つの信号チャネルであり、一つの特定のインタフェースを使用し、これが2つのエンティティの間の通信を可能にする。

SACF規則は、単一のアソシエーション（結合）がある場合に適用し、MACF規則は数個の関係するアソシエーションがある場合に適用し、一つのアソシエーションは一つの信号チャネルであり、一つの特定のインタフェースを使用し、これが2つのエンティティの間の通信を可能にする。

- 5 不幸にして、SACF規則及びMACF規則の全部を単一のFSMで正式に記述するには、一つの過度に複雑なFSMが要求される。この結果は、全部のMACF手順を含む多くの手順は依然として自然言語で定義される。この様式の手順、即ち、アソシエーション当たり単一のFSMにより特定出来ない手順に対して規則を確認する仕組みはない。

- 10 本発明は、それら規則を多数のFSMにより特定することを認めることにより手順を確認する方法に関する。具体的には、各々が一つの特定のFSMを走行させている数個のプロセスが、単一の事象の確認において含まれることが許される。

一つの事象の確認は、2つのフェーズで行われる。第一に、特定の判定基準を満たすかどうかを評価することにより関連するプロセスが選択される。第二の処

- 15 理フェーズにおいて、選択されたプロセスの全部がその事象を処理する。

これは、一つの事象がプロセスのどれかにより処理される前に、関係するプロセスの全部が一つの安定した状態にあるという利点がある。

本発明をより良く理解するため、例として、付随する図面を参照する。

- 20 図1は、一つのコンテナ（container）インタフェースを共有する2つのFSMの表示である。

図2は、Aを走行させているプロセスAxが、Bを走行させている2つのプロセスBx及びByを含む状況を表す。

図3は、それ自身が幾つかの他のFSMから成る一つのFSMを表す。

図4は、セッションビュー（session-view）のためのFSMを表す。

- 25 図5は、呼出しビュー（call-view）のためのFSMを表す。

図6は、Cp又はApのためのFSMを表す。

本発明に従えば、手順の確認は、数個のプロセスを含み、この確認の第一のフェーズは、この確認に関係すべきプロセスの選択を含む。

プロセスの選択は、それぞれのエンティティ及びインタフェースを表すFSM

の特性及び相互関係に基づく判定基準を使用する。

- 第一に、F S Mのどれかが「コンテナインタフェース」を共有するかどうか
決定される。図1は、一つのコンテナインタフェースを共有する2つのF S M、
A及びBの表示である。2つのF S Mがこの形式のインタフェースを共有すると
5 きは、一方が他方に含まれると言い、今度は他方が最初のものを含む。図1のこ
の例において、F S M、AはF S M、Bを含み、またF S M、BはF S M、Aに
含まれる。第二のF S Mに含まれる第一のF S Mは、定義により、第二のF S M
を含むどの第三のF S Mにも含まれる。即ち、もしAがBに含まれ、BがCに含
10 まれると、定義によりAはCにも含まれる。Aは間接的にCに含まれると言うこ
とが出来。別のことが述べられなければ、ここで用いられる用語「含まれる」
は、直接に及び間接に含まれる両方を意味し、従って2つより多くのF S Mの間
の関係を特定出来る。コンテナインタフェース上のアソシエーション経由するプ
ロセス間の相互作用は、他の形式の相互作用とは異なって取扱われる。

- インタフェースは、もし次のことが両方のF S Mに適用されればコンテナイン
15 タフェースとして分類出来る。

一つのF S Mが1つの他方のF S Mに直接に含まれるだけである。

一つのF S Mがそれ自身に含まれない。

一つのF S Mを走行させているプロセスが、最初のF S Mが直接含まれるF S
Mを走行させている1つのプロセスより多くのアソシエーションを持たない。

図1に戻ると、従って、FSM、Bを走行させているプロセスは、FSM、Aを走行させる最大1つのプロセスとアソシエーションを持つだけである。FSM、Aを走行させているプロセスは、しかしFSM、Bを走行させる数個のプロセスとアソシエーションを持つことが出来る（これらのアソシエーションは従ってFSM、Aを走行させているプロセスにおいて関係付けられている）。A又はBは、
25 他の（コンテナでない）インタフェースを持つことも出来る。図1において、FSM、Bは1つの他のインタフェースI1を持つように示される。

包含関係の詳細は、ツリー（tree）の形式でシステムの中に記憶することが出来、以後FSMツリーと呼ぶ。FSMツリーの根は、どの他のFSMにも含

まれない一つのF S Mである。F S Mツリーの複数の葉は、どの他のF S Mも含まない複数のF S Mである。

- 5 包含の概念は、夫々のプロセスにも適用する：1つのプロセスはもう1つのプロセスに含まれるが、これはもし第一により走行されるF S Mが、第二により走行されるF S Mに含まれるならばである。更に、2つのプロセスは、直接か又は間接かどちらかにより、これら両方が（直接に又は間接に）結合

(a s s o c i a t e d) している一つのプロセスを経由して結合されなければならない。例えば、図2は、Aを走行させているプロセスA xが、Bを走行させている2つのプロセスB x及びB yを含む状況を表す。

- 10 F S Mと同様に、これらの包含関係の詳細はツリーの形式で容易にシステムの中に記憶することが出来、以後プロセスツリーと呼ぶ。プロセスツリーを走査する時、別個のデータ構造が使用出来るか、プロセス間のアソシエーションが使用出来るかのいずれかである。同一のプロセスツリー内にある一組のプロセスは、選択プロセスのための1つの根拠を形成する。

- 15 プロセスのための選択判定基準を定義するのに使用される第二の特性は、「コンテナ文脈 (Container Context)」であり、これは一組の予め定義された値であり、確認されるべき一つに事象の異なる可能な文脈を識別している。同一のアソシエーションを使用する異なる事象は、従って異なる文脈を持つことができる。例えば、C S 1において、幾つかの動作は、その呼出しに
20 与する単一パーティの文脈内にある。

複数プロセスには、一つのコンテナ文脈が割り当てられてもよい。この場合、これが、一つの事象と一つのプロセスの間の関係を識別する。

従って、一つの事象（「現行の事象」）が確認されるべき時には、現行の事象のコンテナ文脈に整合する一組のプロセスを識別することが可能である。

- 25 コンテナ文脈の割当は、各F S Mに一組のコンテナ文脈、 $F x = \{C C 1, C C 2, \dots, C C n\}$ を割当てることにより実行される。一つのコンテナ文脈は、もし一つのF S Mが、それが含む又はそれが含まれる一つのF S Mに既に割当てられていれば、そのF S Mには割当られることは出来ない。このことはコンテナ文脈がF S Mに割当られる時にシステムにより検査されるべきである。一つの

事象には単一のコンテナ文脈が割り当てられることが出来る。このコンテナ文脈は、次に現行事象にも割り当てられ、「現行コンテナ文脈」と呼ばれる。あるインタフェース上に到着出来る一組の事象が与えられると仮定すれば、その一組の事象のための全部の可能なコンテナ文脈は、そのインタフェースを共有するFSMを含む

- 5 又はそれに含まれる一つのFSMに割り当てられたに違いない。

プロセスツリーにおける各プロセスには、Fxから一つのコンテナ文脈が割り当てられ、これにより同一FSMを走行させる夫々異なるプロセス、及び同じプロセスツリーの一部には、夫々異なるコンテナ文脈が割り当てられる。

- 例えば、一つのプロトコルにおいて、各々がコンテナ文脈として使用できる30個の可能なレグーid (leg-ids) があるかも知れない。幾つかの動作は、全体の呼出し (例えば、解除呼出し (ReleaseCall)) に適用し、これらはその呼出しをコンテナ文脈として持つであろう。パラメタとして一つのレグーidを持つことの出来る一つの動作は、30の別個の事象 (各可能なレグーidに対して1つ) として見える。これは、その動作を30の可能性ある事象の1つへマップするためにその動作の解析を必要とする (即ち、レグーidパラメタ、又は省略 (default) レグーid値、が動作から事象へのマッピングを決定する)。
- 10
15

- プロセスに対する選択判定基準を定義するのに使用される第三の特性は、「コンテナ作用 (Container Action)」であり、これは一つのFSMにおける状態移行において特定できる特別な一組の作用である。これらの作用は、他のプロセスによる現行事象の処理に影響するかも知れない。例えば、これらは、もう1つのプロセスが、たとえそうする様に指定されとしてもその事象を処理しないようにさせる。
- 20

- 各システムは、定義された幾つかのコンテナ作用を持つであろう。コンテナ作用は他のプロセスに影響することがあるので、数個のプロセスが、相互に影響を及ぼす複数のコンテナ作用を実行する時は衝突が生じるかも知れない。システムは、これらのコンテナ作用の間の特定された明瞭な相互作用を持たねばならない。例えば、より高い優先を持つコンテナ作用は、より低い優先 (現行事象に対して) を持つ全部のコンテナ作用を排除する。これらの相互作用を定義するために
- 25

用いられる原理は、ここではさらに説明しない、システムは、一組のコンテナ作用が与えられると、許されるであろう小組 (subset) のコンテナ作用を識別することが出来ると仮定する。この小組は、以後現行「コンテナ作用」と呼ぶ。

- 5 システムが、異なるプロセスからの異なるコンテナ作用を調整出来るために、一つの事象の処理に先立って2つの段階が必要である。

第一に、その事象を処理できる一組のプロセスが与えられると、これらのプロセスが、一旦これらがその事象を処理すると実行するであろうコンテナ作用が収集される。これは、関係するプロセスによる一つの事象の予備処理が必要である：
10 結果として生じる状態移行が決定され（これらは遂行されるであろう作用を決定するので）、しかし遂行はされない。

第二に、収集されたコンテナ作用、及びシステムにおいてそれらの間に定義された優先に基づいて、システムは、現行コンテナ作用を創作する。

- 一つ的事象が一組のプロセスにより処理されるべき時には、同じ作用が2回実行されるのを回避するため、それらプロセスのコンテナ作用は調整されなければならない。
15

上の述べた様に、確認の第一のフェーズは、上述の特性に基づく判定基準を使用して、確認に関係するプロセスを選択することが含まれる。

- たとえその事象をアソシエーションを経由して初期に受取るプロセスがあるとしても、その事象はそのプロセスによって直ちに処理はされないであろう。その受取るプロセスは、選択フェーズの結果として常に選択されるであろうが、それはそのプロセスがその事象を処理してよいとして選択された時だけである。
20

- この選択フェーズの一部として、現行コンテナ文脈があるかどうか決定される。もし現行コンテナ文脈がなければ、そのときはその事象を受取ったプロセスだけが選択される。もし現行コンテナ文脈があれば、そのときは2つの次の判定基準を満たすいかなるプロセスも選択されるであろう。第一は、そのプロセスは、コンテナ文脈を保持する一つのプロセスを保持するか、包含し、又はコンテナ文脈を保持する一つのプロセスに含まれるかのいずれかでなければならない、このことは入れ代わって、その事象を受け取ったプロセスを包含するか、又はそれに含
25

まれなければならない。第二に、そのプロセスは、現行コンテナ作用の1つ以外にどのコンテナ作用も実行してはならない。

この選択フェーズは、その事象を受取ったプロセスを検査することにより開始する。そこから、プロセスツリーは、根の方へ及び葉の方へ走査される。走査されるあらゆるプロセスに対して、それがコンテナ文脈に整合するかどうか調べられる。各整合するプロセスは選択に加えられる。幾つかの最適化を適用する。

第一に、もし整合するプロセスが、その事象を受取ったプロセスを包含するならば、そのときはいかなる他の整合するプロセスも存在しないであろう。この理由は、もし一つのFSMが、一つのコンテナ文脈が割当てられた一つのFSMを既に包含するか、又はそれに含まれるならば、そのコンテナ文脈はそのFSMに割当てられることは出来ないからである。第二に、もし一つの整合するプロセスが発見されていると、そのときはその同一FSMを走行させるいかなる他の整合するプロセスも存在しないであろう。

これは、少なくとも1つの整合するプロセスを与え、また、発見されたあらゆる整合するプロセスに対して、プロセスツリーは、根の方へ及び葉の方へ再び走査される。各整合するプロセスは、それが前に選択されていなければ、今や選択に加えられる。

次に、前述のように、その事象は、選択されたプロセスの全部により予備処理される。これは、現行コンテナ作用を生じる結果となり、これは代わって選択における最終の組のプロセスを決定する。

選択プロセスが正しく動くためには、あたかも完全なプロセスツリーが最初の事象が確認される前に利用出来るように見えなければならない。このことは、あらゆるFSMに対して、それに割当てられたいかなるコンテナ文脈に対しても一つのプロセスがプロセスツリーに存在しなければならないことを意味する（たとえばそれが状態”アイドル”にあっても）。これが与えることがあるオーバヘッドは、いかなるシステムにとっても受入れられない。次の方法は、要求あり次第プロセスを創作するために使用出来る。

もし、選択プロセス中にツリーの端に到達し（そのプロセスは、今後最後のプロセスと呼ぶ）、しかしこの最後のプロセスにより走行するFSM（今後最後の

F S Mと呼ぶ) が、対応するF S Mツリーの根又は葉でなければ、それなら、

5 ー根の方に走査している時、最後のF S Mが直接に含まれるF S Mに対して1つのプロセスが創作されるが、これは、もしそのF S Mが、現行事象に対してアイドルからの状態移行を特定した一つのF S Mであるか又はこれに含まれる場合であり、「アクション無し (no action)」以外の一つのアクションを伴う。

10 ー葉の方に走査している時、最後のF S Mが直接に含まれる各F S Mに対して1つ又はそれより多くのプロセスが創作され、それは現行事象に対してアイドルからの状態移行を特定した一つのF S Mであるか又はこれ包含し、「アクション無し」以外の一つのアクションを伴う。1つのプロセスがそのF S Mに対して創作されるが、これはもしそのF xが一つの数として現行コンテナ文脈を持つか、又はもしそのF xが空のときである。さもなければ、F S MのF xの各数に対して1つのプロセスが創作される。アソシエーションが次に各プロセスと確立され、この後に追跡が継続する。

15 システムは、もし状態“アイドル”において受取ったならば、各F S Mに対して、どの事象に対して「アクション無し」以外のアクションを特定するかを知らなければならない。

20 それが状態“アイドル”にあり、またそれがいかなる他のプロセスも含まない時は、プロセスは除去されることが出来、またそれが持つコンテナインタフェース上のどのアソシエーションも解除することが出来る。プロセスが状態“アイドル”にはないが、最早いかなるアソシエーションも持たず、即ち、プロセスがぐずぐずしている状況が生じるかも知れない。システムは、これを検出出来るが、なぜならプロセスはその時は根は勿論葉だけであるからである。システムは、その場合特別の事象を発生し、例えば、「孤児にする (orphaned)」であり、これに基づいてプロセスはアイドルに行くべきである（そうでなければ、それは誤りと考えられる）。

25 選択フェーズが完了し、また一群のプロセスが選択された後、各選択されたプロセスは、他の選択されたプロセスとは無関係にその事象を処理する。その事象はそれで確認されることが出来る。

複数事象は夫々異なるアソシエーションの上に到着するかも知れず、また一つの引き続く事象は、一つの前の事象が処理されて終うまでは処理出来ないで、待合せ（キューイング）が必要になる。一つの事象は、もし既に複数の事象が待合せしているか、又はもし既に一つの現行事象があれば、待合せさせられる。事象は、事象を受取ったプロセスの根の処理に対して待合せさせられる。どれくらい多くの異なる優先があるかに依存して異なる待合せが適用する。コンテナインタフェースを経由して受取られる事象は、他の様式のインタフェースを経由して受取られる事象よりも高い優先を持つ。従って、これらの事象に対する待合せは、一つの事象がもう1つの待合せから取出されることが出来る前は、空でなければならない。このことは、全部の関係するプロセスに、確認されるべき次の事象が受取られる前に相互に更新する機会を与える。

この発明は、幾つかのFSMにより記述されているINのためのエリクソン（Ericsson）CS1+INAPプロトコルから取った特定の例への応用を参照してより詳細に説明する。これらFSMの1つはSSF-FSMで、これはそれ自身図3に示す様に幾つかのFSMから成る。

従って、セッションビューFSMは、呼出しビューFSMを含み、これは次にCp及びApFSMを含む。このCp及びApFSMは、一つのインタフェースを共有するが、これはコンテナインタフェースではない。

呼出しビューFSMは、呼出し及びSCF（サービス制御機能）と外部インタフェースを持つ。SACF規則及びMACF規則に対して確認されなければならない事象が到着出来るのはこれらのインタフェース上である。

以下のコンテナ作用が識別される（優先順序に提供される）：

バッファ事象。事象は、バッファされる（バッファ機構それ自身はここでは説明されない）。誤り手順は、これに続かない。

拒否事象。事象は、拒否される。誤り手順が、これに続く。

図4は、セッションビューのためのFSMを表し、これはINAPのためのMACF規則を特定する。そこには1つのMACF規則があり、数個のSCFに対する数個のアソシエーションが同時に存在してもよいが、1つだけが呼出しを制御出来る。

出来る。

コンテナ文脈としてセッションビューを持つ動作はない。セッションビューは、全部の他のFSMを含むので、セッションビューは常にあらゆる事象の確認に関係する。

- 5 SCFとの新しいアソシエーションは、セッションビューに対するFSMが状態“アイドル”、又は“監視関係”(Monitoring Relation)にある時だけ許される。それゆえ、セッションビューに対するFSMが状態“監視関係”にあると、動作“初期DP”(InitialDP)(これはSCFとの新しいアソシエーションを開始させる)は拒否される。動作はそれで無視される。

- 10 図5は、呼出しビュープロセスのためのFSMを表し、これはSCF、及び呼出しとのアソシエーションのためのSACF規則を特定する。SCFとのアソシエーション当たり1つの呼出しビュープロセスがある。このFSMは、単一のSCFプロセスから分かるように、一つの呼出しの構成を表す。

- コンテナ文脈として呼出しビューを持つ少数の動作があり、例えば“解除呼
15 出”し(ReleaseCall)である。

- 状態：セットアップ、未定中、及び活動中は、BCSMにおける状態に対応する。セットアップにある間は、SCFは、依然として通常のCCF手順(動作“収集情報”)を使用してユーザから数字を収集することが出来る。一旦未定中になると、最早出来ない(“収集情報”拒否される)。活動中においては、SC
20 Fは、呼出しを経路決め出来ない(“接続”は拒否される)。その呼出しは、呼出しからの一定の事象により未定中へ戻れるだけである。

- 一旦“代理”(Surrogate)に入ると、呼出しを開始したパーティは、その呼出しを放棄している。それでSCFにとっては最早その呼出しを経路決めすることは出来ない(FSMを活動中から未定中へ移行させた呼出しからの事象
25 は、今やアクション無しで同じ状態へ状態移行を生じさせる)。一旦“監視関係”になると、SCFは最早その呼出しを制御出来ないが、その呼出しにおいて起っている事象を監視出来る(全部の制御動作は拒否される)。注目すべきことは、“監視関係”状態に入ると、一つの事象がこのFSMから発生し、セッションビューへ送られる(“監視関係”への状態移行を生じさせるため)。

- 図6は、Cp又はApのためのFSMを表す。CpとApは、全く同一の状態を持つ。一つの呼出しビュー内においては、1つのCpビューだけがあり得るが、数個のApビューがあってもよい。Cp又はApは、Cp又はApが関係する動作のために、SCF、及び呼出しとのアソシエーションのためのSACF規則を
- 5 特定する。

このFSMは、SSFとSCFとの間の同期化手順、及び呼出しパーティと相互作用するための手順を表す（通常のCCF手順を使用する以外）。

- コンテナ文脈としてCp又はApを持つ動作はない。ビューのうちのどちらが含まれるかは、レグ（leg）であるCp及びApに含まれるFSMにより決定
- 10 される。

図3に示す様に、CpとApの間にはインタフェースがあり、これはレグを1つのビューからもう1つへ移動させるのに使用される（事実上、それはレグに対して含まれるプロセスは解除され、新しいものが他方の側で開始されることを意味する）。

- FSMにおける状態“WFI”において、もしユーザが電話を切ると同期化は破られるかも知れないが、Cpプロセス又はApプロセスは、SCFと同期している（事象はCLSMプロセスから到来する）。注目すべきは、それぞれ異なるApプロセスはこの結果一緒に同期しておらず、またCpプロセスと同期しない：同期化は完全な呼出しには適用しない。
- 15

- ユーザ相互作用が進行している間（FSMは状態“ユーザ相互作用WFI”、又は“ユーザ相互作用処理”にある）、全部の動作は、“切断前方向接続”（Disconnect Forward Connection）を除いてCpによりバッファされるであろう。このバッファは、FSMがWFI又は処理に移行して戻る時に解除される。Apは、“解除呼出し”しをバッファするだけで
- 20
- 25 あり、これは局地的に、即ち、他のFSMに影響することなくバッファされるだけである。従って、Apは、これのためにコンテナ作用を使用しない。

FSMが状態“処理”にあると、SSFとSCFは、同時に走行し、即ち、これらは同期化されない。

Cp及びApにおけるレグのための夫々のFSMは、全部の状態移行、作用等

を含み全く同一である。これらのFSMは、アイドル状態と活動状態だけを持つ。これらのFSMは、夫々の動作に対してコンテナ文脈としての役目をする。Cp及びApにおける夫々のレグは全く同じであるので、また各々は1つの状態（アイドルの他に）を持つだけであるので、レグプロセスは、現存するプロセスを活動中からアイドルへ移行させ、また新しいもの（ターゲットビューにおける）をアイドルから活動中へ移行させることにより、ApからCpへ「移動」させることが出来、この逆も可能である。

レグの位置は、従って、Cpビュー、又はApビューが含まれようとしているかどうかを決定する。「レグ」の概念は、呼出しにおける単一のパーティに割り
10 られるネットワーク資源を表す。Apにおいては1つのレグだけがあり得るが、Cpにおいては数個のレグがあっても良い。

これに基づいて、確認中に一つの事象を処理すべき複数プロセスの選択に使用される判定基準を決定することが可能である。

注目すべきは、コンテナインタフェースの概念は、一般に、外部トリガ（即ち、
15 他のシステムから来たり、同じシステムの一部から来たりする）に対する一つのシステム、又は一つのシステムの一部の行動を説明するのに使用されることである。

請求の範囲

1. インテリジェントネットワークにおける一つの事象を確認する方法であって複数のプロセスを含み、当該方法は、
 - 5 全部のプロセスから予め定められた判定基準を満たすプロセスを選択し、
選択されたプロセスの全部において独立して当該事象を処理し、
当該選択する段階は、プロセス間のコンテナインタフェースを定義することを含み、プロセスを表す有限状態マシンの間の関係を定義することを包含することを特徴とするインテリジェントネットワークにおける一つの事象を確認する方法。
- 10 2. 請求項1に記載の方法において、当該予め定めた判定基準は、
有限状態マシンのいずれかがコンテナインタフェースを共有するかどうかを決定し、
プロセスのいずれかが現行事象のコンテナ文脈に整合するかどうかを決定することを含む、インテリジェントネットワークにおける一つの事象を確認する方法。
- 15 3. 請求項1又は2に記載の方法において、当該事象は最初のプロセスにおいて受取られ、また当該方法は、選択段階が完了するまで最初のプロセスにおける事象の処理を遅延させる、ことを含むインテリジェントネットワークにおける一つの事象を確認する方法。
4. 請求項1、2又は3に記載の方法において、予め定められた判定基準を満たすプロセスを選択する段階は、
 - 20 当該組の判定基準を満たす第1の組のプロセスを選択し、
当該事象を第1の組のプロセスにおいて予備処理し、
予備処理の結果に基づいて第2の組のプロセスを選択する、ことを含むインテリジェントネットワークにおける一つの事象を確認する方法。
- 25 5. 請求項4に記載の方法において、第2の組のプロセスは、これらが現行コンテナ作用を含むときのみ選択される、インテリジェントネットワークにおける一つの事象を確認する方法。
6. 先行する請求項のいずれかに記載の方法において、当該事象を処理する段階は、前に受取った事象が処理されて終うまでは、受取った事象を待合せさせる、

ことを含むインテリジェントネットワークにおける一つの事象を確認する方法。