

「異種測位技術間
共通処理基盤」
仕様書

第 1.0 版

2010 年 3 月 10 日

(財) ニューメディア開発協会

目次

1. 序文.....	4
1.1. 背景と定義	4
1.2. 適応範囲.....	5
1.3. 適応性	5
2. 用語および略語	6
3. アーキテクチャ	8
3.1. 概要.....	8
3.2. モジュール詳細	9
3.2.1. 測位手段モジュール.....	9
3.2.2. 測位手段コントローラモジュール	9
3.2.3. 測位コントローラモジュール	10
3.2.4. 位置補正モジュール.....	11
3.2.5. 測位結果履歴モジュール.....	11
3.2.6. 位置表現変換モジュール.....	12
3.2.7. アクセスコントロールモジュール	12
3.2.8. リクエストコントローラモジュール.....	12
3.2.9. 位置取得 API	13
3.2.10. 位置表現変換 API.....	13
3.2.11. コントロール API.....	13
3.3. 現在位置取得処理.....	14
3.3.1. 現在位置取得処理概要	14
3.3.2. 適応的測位手段選択.....	15
3.3.3. 現在位置取得処理中位置表現変換	18
3.3.4. 測位結果履歴利用	18
3.3.5. 測位手段問い合わせモデル.....	19
3.4. 位置表現変換処理.....	20
3.5. コントロール処理.....	21
3.6. 測位手段間位置補正	22
3.7. 適応的測位手段選択	26
3.8. 測位結果履歴管理および軌跡生成.....	27
3.9. 測位手段抽象化	27
3.10. センサ情報の取り扱い	28

3.11.	測位結果へのフィードバック.....	29
4.	データ構造.....	30
4.1.	位置表現.....	30
4.2.	測位手段プロファイル.....	31
4.3.	測位結果.....	32
4.4.	軌跡.....	32
4.5.	測位モード.....	33
4.6.	イベント.....	33
4.7.	測位手段状態.....	34
4.8.	バージョン.....	34
5.	アプリケーションインタフェース.....	34
5.1.	位置取得 API.....	35
5.2.	位置表現変換 API.....	36
5.3.	コントロール API.....	37
6.	測位手段コントローラインタフェース.....	39

1. 序文

1.1. 背景と定義

異種測位技術間共通処理基盤とは、複数の異なる測位技術間で相互に情報連携を可能とすることで測位精度や利便性向上を目的としたプラットフォームである。

近年、GPS のみならず、Wi-Fi アクセスポイントや携帯電話基地局を利用した測位技術が携帯情報端末等に標準搭載され、位置情報を活用したサービスが爆発的に普及しつつある。また、RF-ID や可視光線、あるいは画像認識技術を用いたものなど、多様な特性をもった測位技術の研究開発が進められている。一方で、単一測位技術のみであらゆる状況下で高品質な測位を行うことは一般的に困難であるため、複数測位技術を状況に応じてシームレスに切り替える技術の確立が重要な課題となっている。

図 1(a)に示すように、従来、位置情報を利用したアプリケーションは測位手段を直接利用している。これは、測位手段の利用方法や得られる位置情報の表現方法は測位手段ごとによって異なり、アプリケーションレベルで複数の測位手段インタフェースに対応する必要があり、煩雑である。位置補正機能や移動履歴(軌跡)管理機能などもアプリケーションごとに用意され、冗長である。また、測位手段によってはマルチアクセスに対応していないものもあり、一度に1つのアプリケーションのみしか測位手段を利用できない場合もある。

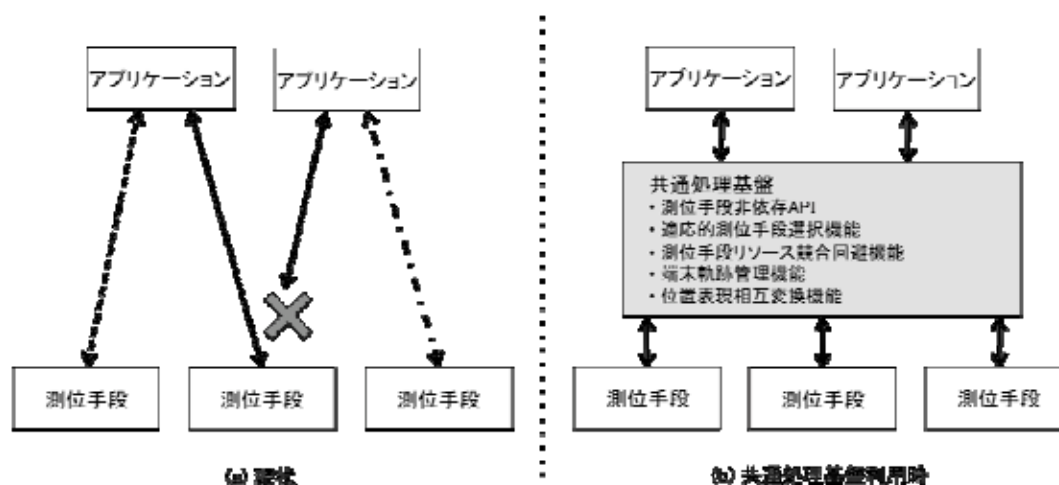


図 1： 共通処理基盤の役割

共通処理基盤では、これらの課題を解決するものとして、次の機能を実現するものとする(図 1(b))。

-
- 測位手段非依存 API
 - 利用している測位手段に依存しない統一的な位置取得アプリケーションインタフェース
 - 適応的測位手段選択機能
 - 端末の状態やアプリケーションの要求に応じて最適な測位手段を選択する
 - 測位手段をシームレスに切り替える
 - 測位手段リソース競合回避機能
 - 複数のアプリケーションから測位手段へのアクセスがあった場合の競合を回避する
 - 端末軌跡管理機能
 - 複数の測位手段から得られる位置情報を時系列による軌跡として扱う
 - 位置表現相互変換機能
 - アプリケーションの要求に応じて、位置表現を変換する。

1.2. 適応範囲

異種測位技術間共通処理基盤は「異種測位技術間における共通処理基盤構築事業 調査報告書第4章4.3要件定義、仕様(1)要件定義」にて規定されている異種測位技術間共通処理基盤に対する要件を満たすものであり、本仕様書ではその概念的な構造を規定する。

1.3. 適応性

本仕様の実装システムが適合するためには、次の条件を満たさなければならない。

- ・ 第3章で規定するモデルに従う
- ・ の抽象概念に従い、データを実装するとき、第4章で規定する情報を含むデータが交換できなければならない。
- ・ の抽象概念に従い、データを交換するとき、第5章および第6章で規定するインタフェースを実装しなければならない。

2. 用語および略語

本仕様書で使用する用語および略語を表 1 に示す。

表 1：用語および略語

用語・略語	説明
共通処理基盤	異種測位技術間共通処理基盤のこと。
測位手段	GPS や屋内 GPS[1]、PlaceEngine などの位置情報取得機能を提供する機構のこと。
測位モード	より精度が高い位置情報が欲しい、即座に結果が欲しいなどといったアプリケーションの要求を抽象化したもので、共通処理基盤の動作を大まかに規定するもの。
測位結果	測位によって得られた位置情報に加え、測位時刻や測位精度といったある測位行為に付随した情報のこと。
測位時刻	測位結果を得た時刻のこと。ある測位行為に付随した情報の一つで、位置情報と対にすることで端末の 4 次元的な時空間情報を表す。
測位精度	ある測位結果における位置情報の最大誤差を表すパラメタのこと。
測位結果信頼度	共通処理基盤が測位結果をどの程度担保できるかを表すパラメタのこと。
位置情報	場所を表す情報のことで、参照する空間によって表現がことなる。例えば、世界測地系を参照するなら、緯度経度高度で表現される。
位置表現変換	ある位置情報を別の空間の位置情報にマッピング、変換すること。
センサ	地磁気センサや加速度センサといった、位置情報以外の計測値を提供する手段のこと。
測位結果履歴	指定された期間の過去における測位結果集合のこと。軌跡と異なり、未加工の測位結果の集合を指す。
軌跡	指定された期間における位置情報のこと。測位結果履歴と異なり、空間の統一と内挿などの加工が施された位置情報の集合を指す。
位置補正	観測された位置情報の誤差を修正する行為のこと。位置補正は A-GPS のような測位手段に依存のものと、マップマッチングのような測位手段に依存しないものがある。

測位手段間位置補正	測位手段非依存位置補正のひとつで、他の測位手段の測位結果をもって、位置補正を行う行為のこと。例えば、測位手段が切り変わる際に測位結果のばらつきを補正する行為が挙げられる。
位置表現	空間識別子と位置情報の対のこと。詳細は第 4.1 節に示す
同期/非同期 API	API を呼び出した際に、要求された処理がすべて終わるまで返答しない方法を同期 API、処理の完了を待たずにすぐに呼び出し元に制御が戻る方法を非同期 API と呼ぶ。非同期 API では、処理の結果は、他の呼び出しを通じて返答される。
PlaceEngine	Wi-Fi 機器を使って簡単に現在位置を推定するサービスのひとつ。
屋内 GPS[1]	GPS と同等の無線信号を利用し、送信機から位置情報そのものを放送することで屋内測位を実現する手段のひとつ。GPS 受信機のハードウェアをそのまま利用できる。

3. アーキテクチャ

3.1. 概要

共通処理基盤は、次のコンポーネントで構成されるものとする。

- 共通処理基盤は測位手段間の差異を吸収する中間レイヤをもち、このモジュール性が測位手段に依存しないインタフェースを提供する。
- 最適な測位手段を判断するために、共通処理基盤は測位手段の特徴を記述したプロファイルおよびアプリケーションのポリシーを管理し、測位手段や端末の状態を適宜監視する。
- 測位手段へのアクセスはアプリケーション毎ではなく共通処理基盤が一括して行い、アプリケーション間の測位手段アクセス競合を防ぐ。
- 測位結果は、ある一定期間、共通処理基盤内で記憶する。この履歴は短期的なものと長期的なものに分類する。短期記憶は、パフォーマンスの向上やリソース競合回避のために利用され、長期記憶は端末の軌跡管理にも利用される。
- 共通処理基盤はある空間と別の空間との間の位置情報のマッピングルールを保持することで、位置表現の相互変換機能を実現する。

図 2 は、上記のようなコンポーネントの関係を示している。図中、角丸四角は共通処理基盤内のモジュール、四角は基盤外のモジュールを表している。また、矢印は API 利用時のデータフローを表し、線矢印は位置情報取得時のフロー、点線矢印は位置表現変換時のフロー、破線矢印は共通処理基盤を設定管理する際のフローを表している。

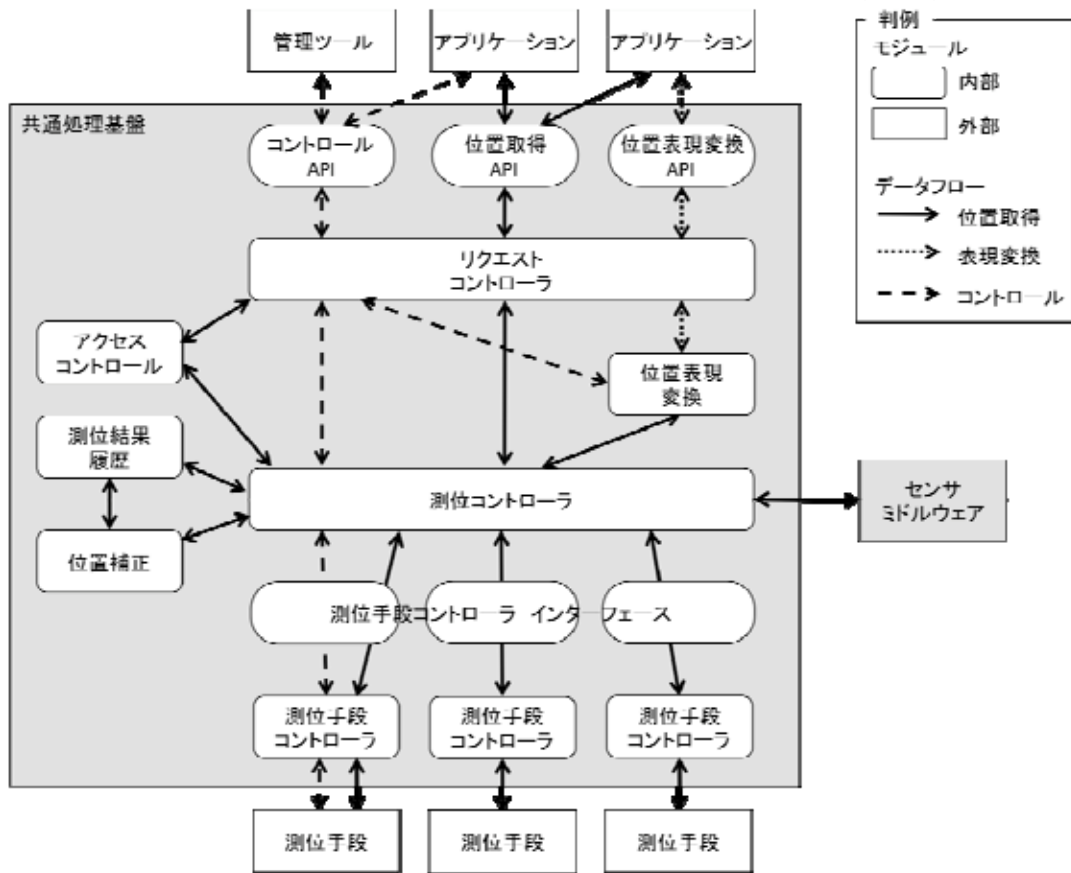


図 2：共通処理基盤アーキテクチャ

各モジュールの詳細は、次節以降で説明する。

3.2. モジュール詳細

3.2.1. 測位手段モジュール

測位手段モジュールは、プリミティブな位置情報を取得する機構そのものと、それを管理する機構である。例えば、GPS の場合は、GPS デバイスおよびデバイスドライバが測位手段モジュールに該当する。

当モジュールは、測位機能や位置補正機能に必要な情報をやりとりする管理機能も有する。

3.2.2. 測位手段コントローラモジュール

測位手段コントローラモジュールは、測位手段間の差異を吸収し、共通処理基盤内の上位モ

ジュールに対して統一的なインタフェースを提供するモジュールである。例えば、Window 7 の Location & Sensor API の Location Provider のような役割を果たす。

当モジュールは、測位手段の特性を示すプロファイルを管理・提供する機能も有する。上位モジュールが測位手段を選択する際に、このプロファイルが参照される。

新しい測位手段を共通処理基盤に組み込む際には、測位手段提供者は当モジュールを用意するだけで良い。

3.2.3. 測位コントローラモジュール

測位コントローラモジュールは、リクエストコントローラからの要求により、測位手段コントローラから位置情報を取得する機能を有する。リクエストに応じて位置表現を変換するため、当モジュールは位置表現変換モジュールと連携する。

測位手段が明示的に選択されていないリクエストの場合、最適な測位手段を選択する機能も有する。この選択は、アプリケーションのポリシーや測位手段プロファイル、端末の状態などを考慮して行われる。

測位コントローラ内では、測位手段コントローラから取得できた位置表現空間のままで記憶され、必要に応じて変換処理を行うものとする。これは、他の空間にマッピングできない空間もあり、測位コントローラ内で標準的な空間を定義できないためである。

当モジュールは、測位手段を管理する機能も有する。例えば、消費電力の節約のため、利用していない測位モジュールの電源を切るなどの処理を行う。具体的なアルゴリズムは、競争になる部分とし、本仕様書では規定しない。

図 3 に、測位コントローラのモデルを示す。リクエストの受付あるいはタイマの期限切れ/イベントの発生により、実処理を行うワーカを生成し、ワーカが適宜必要なモジュールと連携して位置取得機能を実現する。

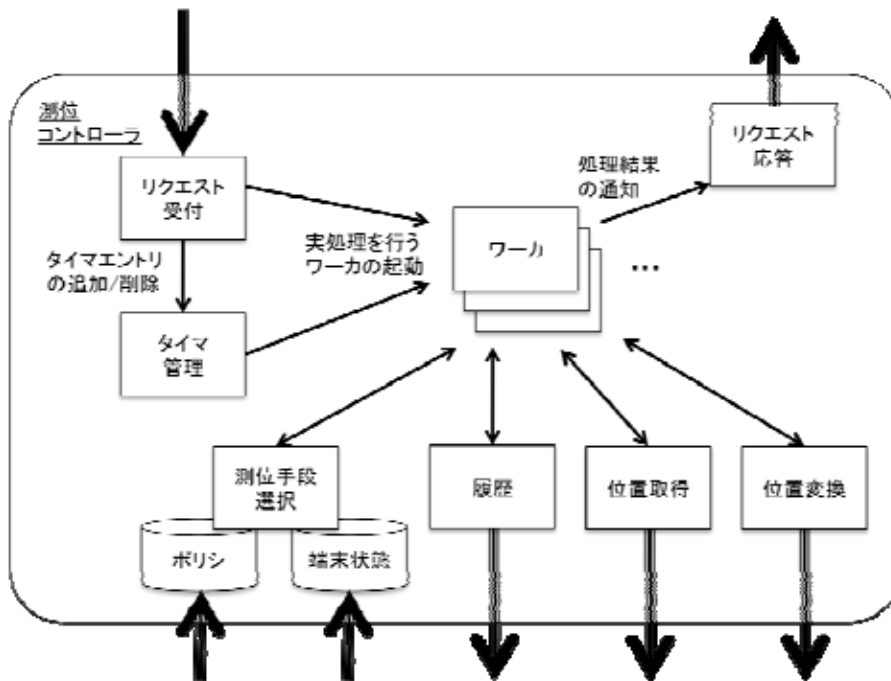


図 3：測位コントローラ

3.2.4. 位置補正モジュール

位置補正モジュールは、測位手段に依存しない位置補正機能や、測位手段間で行う位置補正の機能を有する。例えば、マップマッチングや他測位手段の位置情報を用いたはずれ値判定などを行う。

共通処理基盤は、個々の測位手段から提供されるプリミティブな位置情報を、未加工の形でアプリケーションへ提供する機能も有する。ナビゲーションソフトなどには、独自の高度な位置補正機能を備えているものもあり、これら既存アプリケーションが持つ資産を引き続き活用することや、アプリケーションの独自機能として特徴付けることも可能である。また、既存ノウハウを活用した位置補正機能を共通処理基盤内に実装することにより、アプリケーションに高度な位置補正機能を提供することも可能である。

3.2.5. 測位結果履歴モジュール

測位結果履歴モジュールは、測位手段ごとの測位結果を記憶しておくモジュールである。この履歴は短期履歴と長期履歴から構成される。短期履歴は、パフォーマンスの向上やリソース競合回避のために利用される。長期履歴は端末の軌跡管理に利用される。

短期履歴は、パフォーマンスの向上やリソース競合回避のために利用される。よって、短期履歴は実装上必要最低限な量および期間を保持することし、アプリケーションから参照したり削除したりしないものとする。

長期履歴は端末の軌跡管理に利用される。長期履歴は、プライバシー保護の観点から、セキュアな方法で保持される必要がある。アプリケーションから履歴を参照するインターフェースを用意するが、ユーザの同意なしあるいは意図しない履歴の取得は認めないものとする。また、長期履歴はユーザが任意のタイミングで消去できるものとする。

3.2.6. 位置表現変換モジュール

位置表現変換モジュールは、座標や住所、施設名等、場所を特定するための様々な識別子および位置表現を相互に変換するモジュールである。

3.2.7. アクセスコントロールモジュール

アクセスコントロールモジュールは、あるアプリケーションが要求してきた処理を実行する権利があるかどうかを確認するモジュールである。位置情報を利用する際の Disclaimer (免責事項) 確認機能も本モジュールの役割に該当する。

アクセスコントロールモジュールの判定の結果、要求処理を実行することができない場合は、アクセスコントロールが原因で要求が失敗している旨をアプリケーションに通知する必要がある。

3.2.8. リクエストコントローラモジュール

リクエストコントローラは、アプリケーションからの要求を受け付けて、共通処理基盤内の適切な処理/機能を呼び出すモジュールである。

非同期呼び出しの場合は、コールバックを必要とするリスナを管理し、タイマやイベントの生成・割当を行う。

図 4 に、リクエストコントローラのモデルを示す。リクエストの受付あるいはタイマの期限切れ/イベントの発生により、実処理を行うワーカを生成し、ワーカが適宜必要なモジュールと連携して要求された処理を実行する。

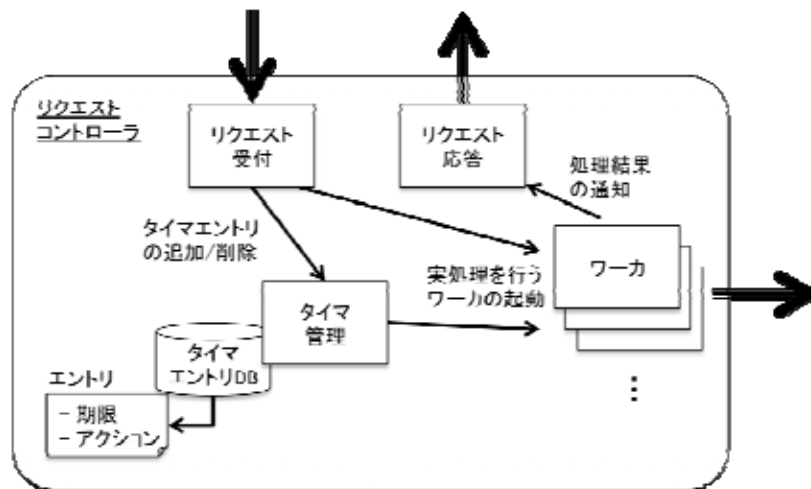


図 4: リクエストコントローラ

具体的なアルゴリズムは、競争になる部分とし、本仕様書では規定しない。

3.2.9. 位置取得 API

位置取得 API は、アプリケーションが共通処理基盤から位置情報の提供を受けるための手続きを定めた規約の集合である。アプリケーションは、位置取得 API に則り、測位や位置情報取得に係るリクエストや各種の設定変更ができる。

個々の API の詳細については第位置取得 API5.1 節に記す。

3.2.10. 位置表現変換 API

位置表現変換 API は、アプリケーションが共通処理基盤に位置表現を引き渡し、異なる空間の位置表現の提供を受けるための手続きを定めた規約の集合である。アプリケーションは、位置表現変換 API に則り、位置表現の引渡しや希望する空間の指定ができる。

個々の API の詳細については第 5.2 節に記す。

3.2.11. コントロール API

コントロール API は、アプリケーションあるいは管理ツールが共通処理基盤を管理するための手続きを定めた規約の集合である。アプリケーションあるいは管理ツールは、コント

ロール API に則り、例えば利用可能な測位手段のリスト取得や、測位手段の測位間隔の変更などができる。

個々の API の詳細については第 5.3 節に記す。

3.3. 現在位置取得処理

本節では、共通処理基盤における現在位置取得処理の流れを説明する。

3.3.1. 現在位置取得処理概要

非同期な呼び出し方法で現在位置取得処理を行う場合を例に、現在位置取得処理時の共通処理基盤の振る舞いを説明する。その処理の流れを図 5 に示す。

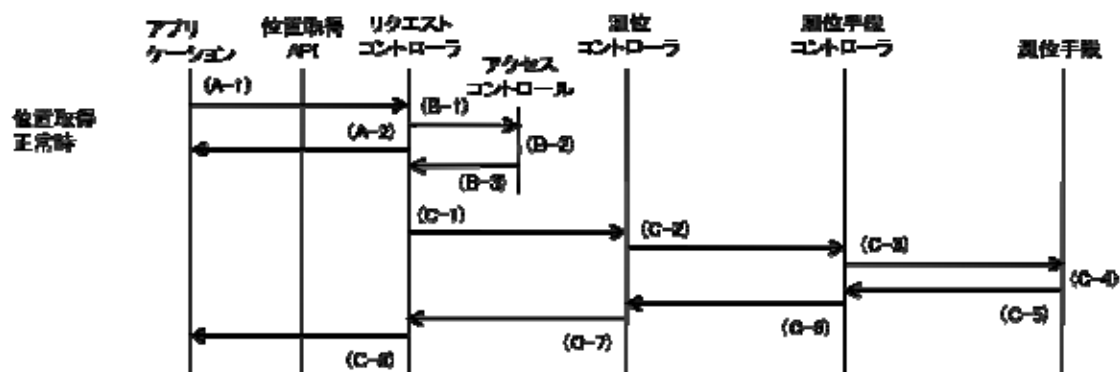


図 5：現在位置取得処理の流れ

(A-1)非同期 API で現在位置取得を要求する。

(A-2)正常に要求を受けた場合は、即座に返答する。

(B-1)要求を受けたアプリケーションが位置情報を利用していいかを確認する。

(B-2)アプリケーションが既に利用許可が得られているかを確認する。初めて位置情報を利用しようとしている場合もしくは以前に利用許可が得られていなかった場合は、ユーザに利用許諾許可を伺う。

(B-3)ここでは、既に利用許可が得られている場合もしくは許可が得られた場合とし、利用許可を通知する。

(C-1)現在位置取得処理を開始する。測位コントローラに現在位置取得を要求する。

(C-2)測位コントローラは、測位手段コントローラに現在位置取得を要求する。

- (C-3)測位手段コントローラは、測位手段に現在位置取得を要求する。
- (C-4)測位手段は現在位置を取得する。
- (C-5)測位結果を測位手段コントローラに返答する。
- (C-6)測位結果を測位コントローラに返答する。
- (C-7)測位結果をリクエストコントローラに返答する。
- (C-8)測位結果をアプリケーションにコールバック返答する。

アプリケーションが現在位置取得処理を許可されない場合の処理の流れを図 6 に示す。

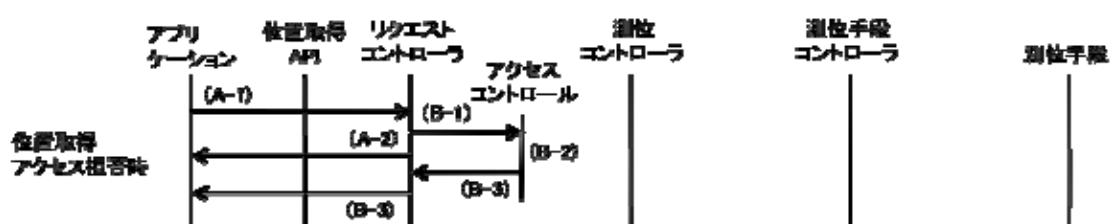


図 6：現在位置取得処理拒否時の流れ

- (A-1)非同期 API で現在位置取得を要求する。
- (A-2)正常に要求を受けた場合は、即座に返答する。
- (B-1)要求を受けたアプリケーションが位置情報を利用していいかを確認する。
- (B-2)アプリケーションが既に利用許可が得られているかを確認する。初めて位置情報を利用しようとしている場合もしくは以前に利用許可が得られていなかった場合は、ユーザに利用許諾許可を伺う。
- (B-3)ここでは、既に利用許可が得られなかった場合とし、利用不可を通知する(現在位置取得失敗)。

3.3.2. 適応的測位手段選択

測位手段が複数存在する場合、共通処理基盤は端末の状態やユーザのポリシーに応じて適切な測位手段を選択する機能を持つ。適応的測位手段選択処理の流れを図 7 に示す。

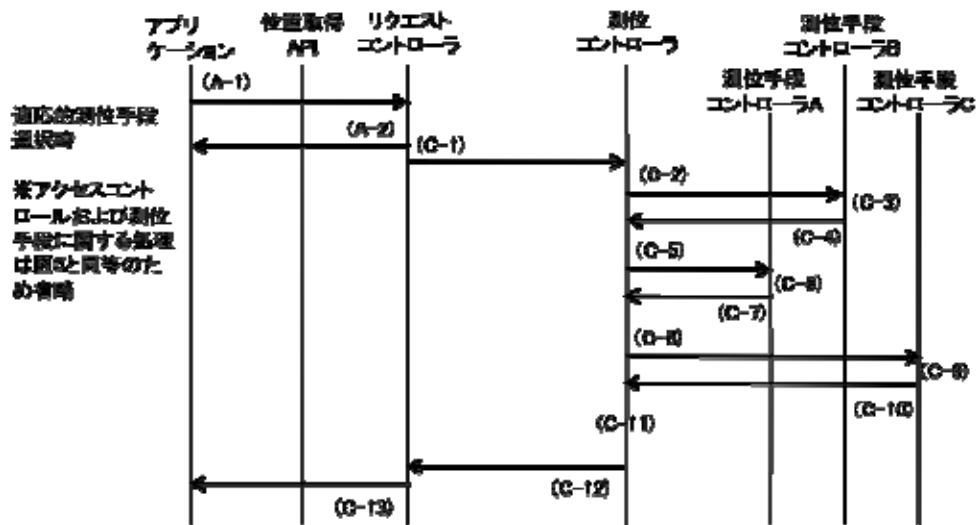


図 7：適応的測位手段選択時処理の流れ 1

(A-1)非同期 API で適応的測位手段選択による現在位置取得を要求する。

(A-2)正常に要求を受けた場合は、即座に返答する。

(C-1)適応的測位手段選択による現在位置取得処理を開始する。測位コントローラに適応的測位手段選択による現在位置取得を要求する。

(C-2)測位コントローラは、個々の測位手段コントローラに現在位置取得を要求する。

(C-3)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け

る。

(C-4)測位結果を測位コントローラに返答する。

(C-5)測位コントローラは、個々の測位手段コントローラに現在位置取得を要求する。

(C-6)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け

る。

(C-7)測位結果を測位コントローラに返答する。

(C-8)測位コントローラは、個々の測位手段コントローラに現在位置取得を要求する。

(C-9)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け

る。

(C-10)測位結果を測位コントローラに返答する。

(C-11)測位コントローラは、全ての測位手段からの測位結果を比較し、別途設定されているポリシーに従って最良とされる単一の測位手段を選択する。

(C-12)選択された測位手段による測位結果をリクエストコントローラに返答する。

(C-13)測位結果をアプリケーションにコールバック返答する。

測位手段を選択する具体的なアルゴリズムは競争領域のため本仕様では特に定義しないが、

上記の例のようにすべての測位手段コントローラに現在位置問い合わせを行い、その結果のなかから最適な測位結果を選ぶことは、一般に高コストである。測位コントローラが測位手段プロファイルなどから必要であると判断した測位手段のみに現在位置問い合わせを行うことも、適応的測位手段選択と言える。このような処理の流れを図 8 に示す。

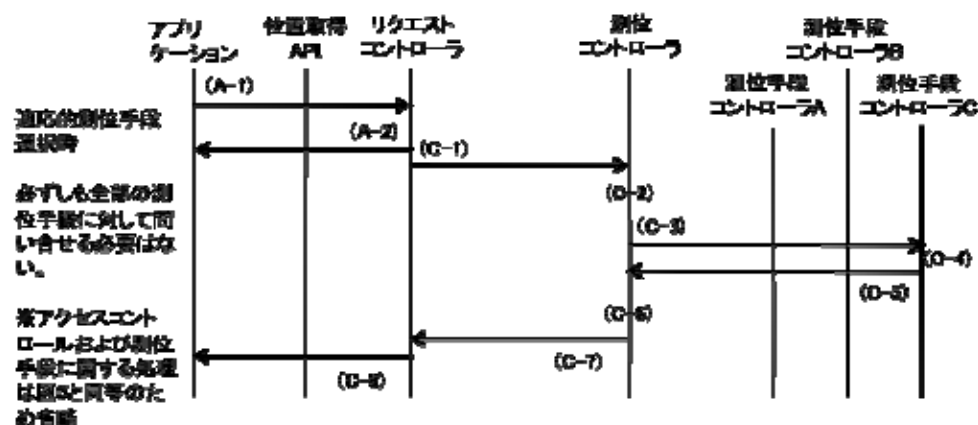


図 8：適応的測位手段選択時処理の流れ 2

(A-1)非同期 API で適応的測位手段選択による現在位置取得を要求する。

(A-2)正常に要求を受けた場合は、即座に返答する。

(C-1)適応的測位手段選択による現在位置取得処理を開始する。測位コントローラに適応的測位手段選択による現在位置取得を要求する。

(C-2)測位コントローラは、全ての測位手段の測位手段プロファイルを比較し、別途設定されているポリシーに従って必要とされる測位手段を一つ以上選択する。

(C-3)測位コントローラは、選択された個々の測位手段コントローラに現在位置取得を要求する。

(C-4)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け取る。

(C-5)測位結果を測位コントローラに返答する。

(C-6)測位コントローラは、選択された全ての測位手段からの測位結果を比較し、別途設定されているポリシーに従って最良とされる単一の測位結果を選択する。

(C-7)選択された測位手段による測位結果をリクエストコントローラに返答する。

(C-8)測位結果をアプリケーションにコールバック返答する。

3.3.3. 現在位置取得処理中位置表現変換

アプリケーションが指定した空間識別子と測位手段から得られた測位結果と対になる空間識別子が異なる場合、共通処理基盤は測位結果を指定された空間に対応した位置表現に変換する必要がある。位置表現変換を伴う現在位置取得処理の流れを図 9 に示す。

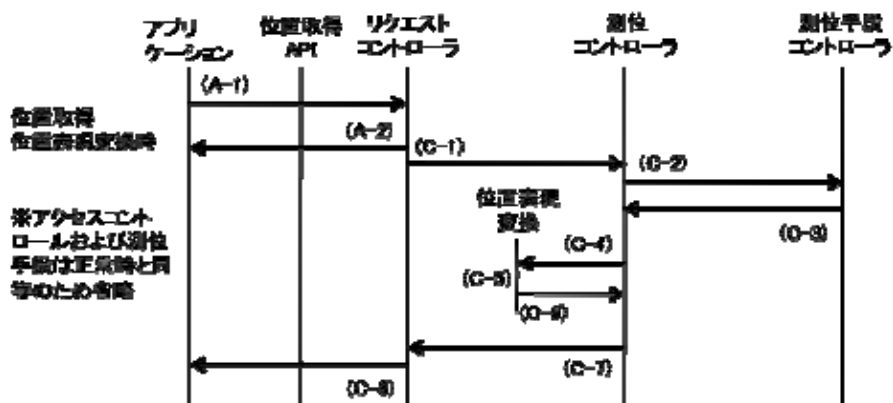


図 9：位置表現変換を伴う現在位置取得処理の流れ

(A-1)非同期 API で現在位置取得を要求する。

(A-2)正常に要求を受けた場合は、即座に返答する。

(C-1)現在位置取得処理を開始する。測位コントローラに現在位置取得を要求する。

(C-2)測位コントローラは、測位手段コントローラに現在位置取得を要求する。

(C-3)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け、測位コントローラに返答する。

(C-4)測位結果と対になる空間識別子が、指定された空間識別子と一致しない場合、測位コントローラは、位置表現変換モジュールに位置表現変換を要求する。

(C-5)位置表現変換モジュールは、位置表現を変換する。

(C-6)位置表現変換された測位結果を測位コントローラに返答する。

(C-7)位置表現変換された測位結果をリクエストコントローラに返答する。

(C-8)位置表現変換された測位結果をアプリケーションにコールバック返答する。

3.3.4. 測位結果履歴利用

現在位置取得処理を複数のアプリケーションから要求される場合や位置取得に時間がかかる測位手段を利用する場合等において、測位結果履歴を利用すると負荷軽減や応答性の向上などに有効な場合がある。測位結果履歴を利用した場合の現在位置取得処理の流れを図

10 に示す。

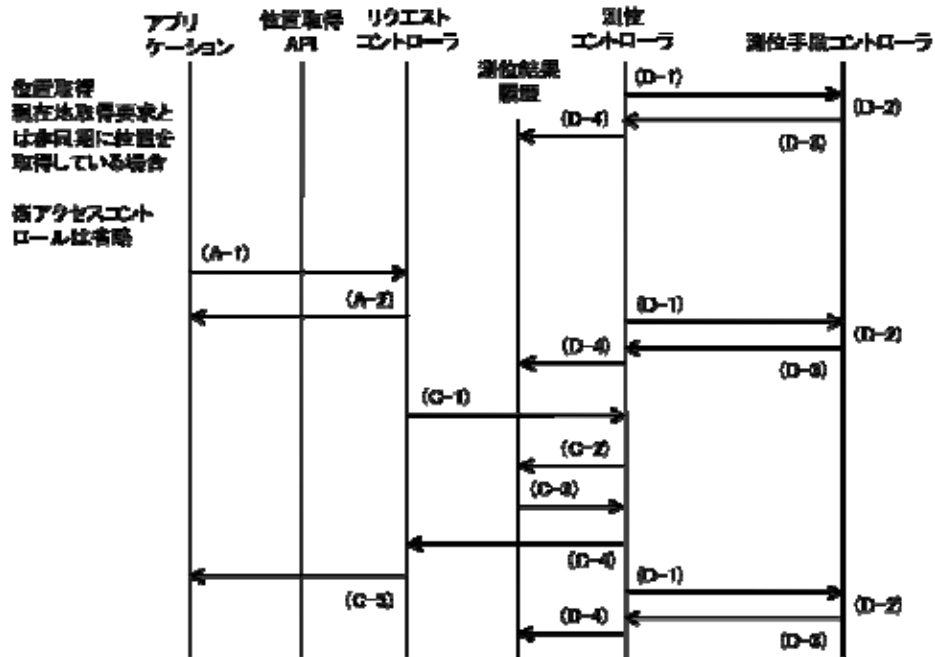


図 10：測位結果履歴を利用した現在位置取得処理の流れ

(A-1)非同期 API で現在位置取得を要求する。

(A-2)正常に要求を受けた場合は、即座に返答する。

(C-1)現在位置取得処理を開始する。測位コントローラに現在位置取得を要求する。

(C-2)測位コントローラは、測位結果履歴に現在位置取得を要求する。

(C-3)短期記憶されている最新の位置情報を測位コントローラに返答する。

(C-4)最新の測位結果を測位コントローラに返答する。

(C-5)最新の測位結果をアプリケーションにコールバック返答する。

(D-1)測位コントローラは、定期的に測位手段コントローラに現在位置取得を要求する。

(D-2)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け
る。

(D-3)測位結果を測位コントローラに返答する。

(D-4)測位結果履歴は、測位コントローラから測位結果を受けとり、記憶する。

3.3.5. 測位手段問い合わせモデル

位置情報を問い合わせる手順は、表 2 に示したように、測位手段ごとに異なる。同期方式は、

現在位置取得要求と応答とが一對一で対応する方式で、PlaceEngine などが該当する。一方、非同期方式は、現在位置取得が一度要求された後に定期あるいは不定期に現在位置が返答されるタイプで、GPS や RF-ID などが該当する。一般的に GPS はシリアルデバイスから 1 秒間隔で定期的に位置情報が取得できるのに対し、RF-ID はタグが読めたタイミングで位置情報が取得できるため、定期および不定期という分類をしている。

表 2：測位結果取得方式

方式	間隔	代表的な測位手段
同期		PlaceEngine
非同期	定期	GPS
	不定期	RF-ID

共通処理基盤では、これらの問い合わせモデルに対応するために、測位手段コントローラが測位手段から測位結果を取得するフローは、図 10 に示す 2 種類を想定している。

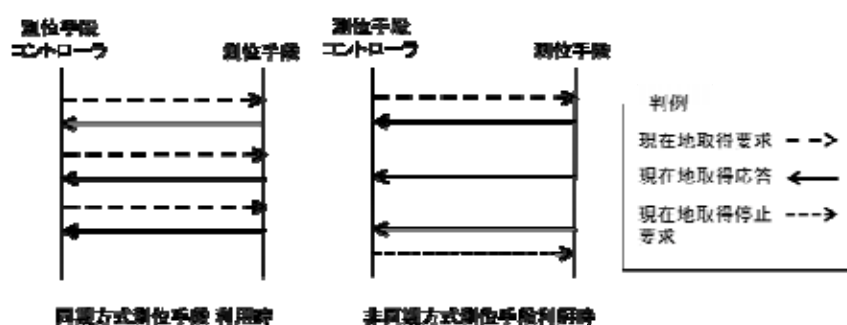


図 11：測位手段問い合わせ処理の流れ

同期方式には、測位手段コントローラは、測位コントローラからの現在位置取得要求やインターバルタイミイベントなどをきっかけとして、測位手段に現在位置取得を要求する。

非同期方式には、測位手段コントローラは、測位手段の利用開始の際に現在位置取得を要求し、測位手段の利用終了時に現在位置取得停止を要求する。このタイプでは、図 10 に示した(D-1)のメッセージは利用開始時に受け渡しされれば十分であり、それ以降の(D-1)メッセージは省略できる。

3.4. 位置表現変換処理

共通処理基盤は、測位機能とは独立して位置表現変換機能を提供する。

位置表現変換処理は、次のように行われる。



図 12：位置表現変換処理の流れ

(A-1)位置表現変換を要求する。

(A-2)リクエストコントローラは、位置表現変換モジュールに位置表現変換を要求する。

(A-3)位置表現変換モジュールは、指定された第一の位置表現（第一の位置情報と第一の空間識別子）を基に、指定された第二の空間識別子に対応する第二の位置情報を生成する。

(A-4)第二の空間識別子と、生成された第二の位置情報によって構成される第二の位置表現を、リクエストコントローラに返答する。

(A-5)第二の位置表現をアプリケーションに返答する。

位置表現変換モジュールでは、例えばある空間のある座標が別の空間のどの座標に該当するかを紐づけるような変換ルールを管理する必要がある。変換ルールは、全国規模の大規模なデータベースを要するものから簡易な定数によるマッピングを行うものまで多岐にわたるルールが想定される。しかし、現時点では、変換ルールの一律的な管理設定方法は確立してない。したがって、位置表現変換ルールを設定を行うインタフェースの必要性を認めつつも、現時点ではそのインタフェースを設けないものとする。

3.5. コントロール処理

共通処理基盤の設定・管理するコントロール処理は、次のように行われる。

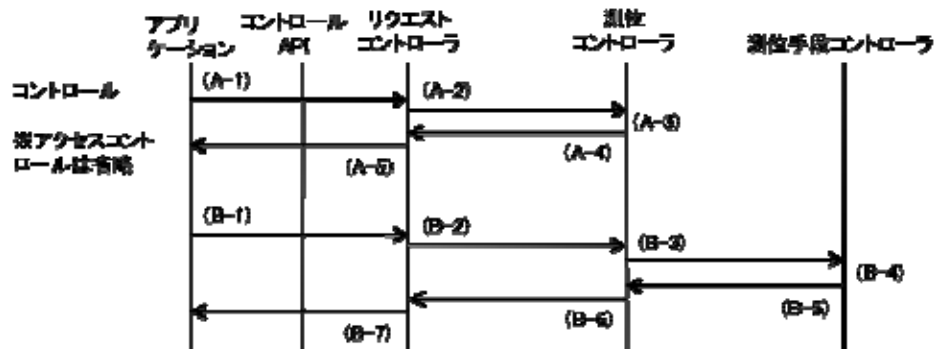


図 13：コントロール処理の流れ

測位モードの変更設定を例に説明する：

- (A-1)測位モードの変更設定を要求する。
- (A-2)リクエストコントローラは、測位コントローラに測位モードの設定を要求する。
- (A-3)測位コントローラは、指定された測位モードに切り替える。
- (A-4)測位コントローラは、測位モードの切り替えに成功した旨を返答する。
- (A-5)測位モードの切り替えに成功した旨をアプリケーションに返答する。

測位手段の測位間隔を設定する処理を例に説明する。

- (B-1)測位手段の測位間隔変更を要求する。
- (B-2)リクエストコントローラは、測位コントローラに測位手段の測位間隔変更を要求する。
- (B-3)測位コントローラは、指定された測位手段コントローラに対して、測位間隔変更を要求する。
- (B-4)測位手段コントローラは、測位間隔を変更する。
- (B-5)測位手段コントローラは、測位間隔の変更が成功した旨を返答する。
- (B-6)測位コントローラは、測位間隔の変更が成功した旨を返答する。
- (B-7)リクエストコントローラは、測位間隔の変更が成功した旨を返答する。

3.6. 測位手段間位置補正

測位手段間位置補正を用いた処理は、次のように行われる。

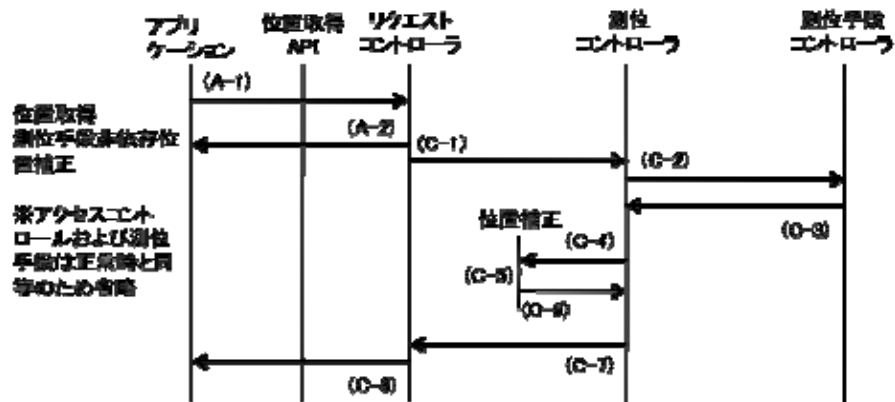


図 14：測位結果位置補正利用時処理の流れ

- (A-1)非同期 API で現在位置取得を要求する。
- (A-2)正常に要求を受けた旨を、即座に返答する。
- (C-1)現在位置取得処理を開始する。測位コントローラに現在位置取得を要求する。
- (C-2)測位コントローラは、測位手段コントローラに現在位置取得を要求する。
- (C-3)位置情報を測位コントローラに返答する。
- (C-4)取得した位置情報を補正にかける。
- (C-5)例えばマップマッチングなどを行う。
- (C-6)補正後の位置情報を返答する。
- (C-7)補正された位置情報をリクエストコントローラに返答する。
- (C-8)補正された位置情報をアプリケーションにコールバック返答する。

測位手段を切り替える際に、測位手段間で測位結果が大きく異なると、ユーザビリティを損なう可能性がある。そこで、測位手段切り替え時には、次のように位置補正モジュールと測位結果履歴モジュールとを組み合わせる。

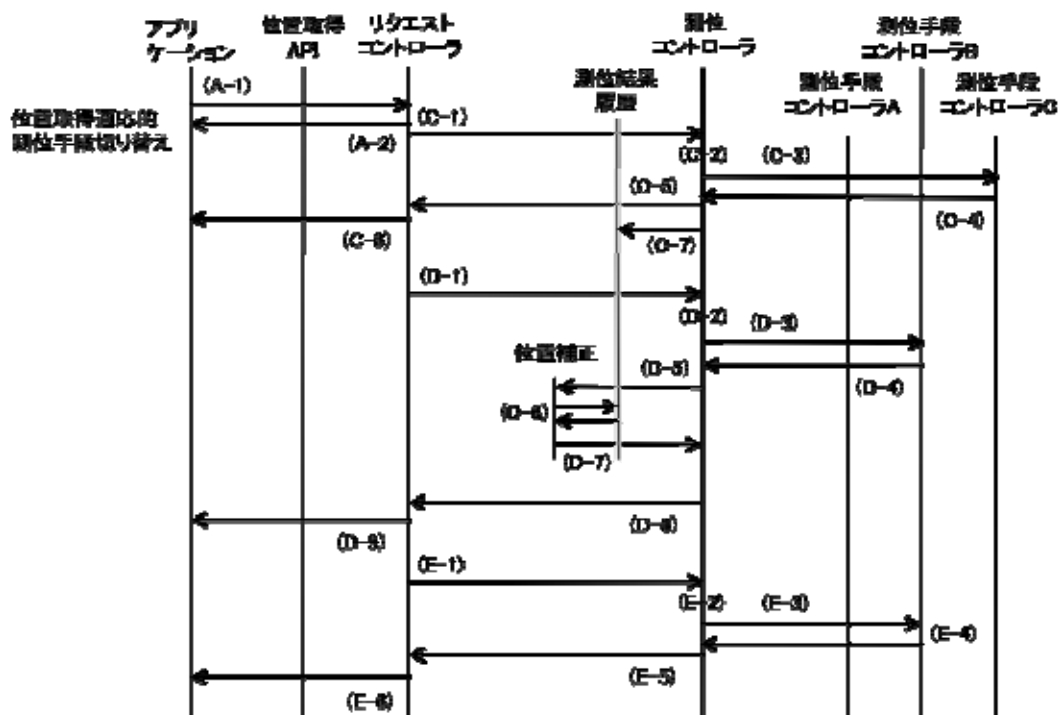


図 15：位置補正利用時処理の流れ(測位手段切替時の補正/補間)

(A-1)非同期 API で定期的な現在位置取得を要求する。

(A-2)正常に要求を受けた旨を、即座に返答する。

(C-1)現在位置取得処理を開始する。測位コントローラに現在位置取得を要求する。

(C-2)測位コントローラは、端末の状態やユーザの設定を鑑みて、測位手段 C を利用することを決める。

(C-3)測位手段 C のコントローラに現在位置取得要求を行う。

(C-4)測位手段コントローラは、測位コントローラに測位結果を返答する。

(C-5)測位コントローラは、リクエストコントローラに測位結果を返答する。

(C-6)リクエストコントローラは、アプリケーションに測位結果を返答する。

(C-7)測位コントローラは、この測位結果を記録する。

(D-1)指定された間隔が経過した後、リクエストコントローラは測位コントローラに再び現在位置取得を要求する。

(D-2)測位コントローラは、端末の状態やユーザの設定を鑑みて、測位手段 B を利用することを決める。

(D-3)測位手段 B のコントローラに現在位置取得要求を行う。

(D-4)測位手段コントローラは、測位コントローラに測位結果を返答する。

(D-5)測位コントローラは、測位手段切り替え時に生じる可能性がある測位結果のギャップ

を緩和するために、位置補正モジュールに問い合わせる。

(D-6)位置補正モジュールは、直前の測位結果もしくは測位結果の履歴を参照して、はずれ値判定やスムージング処理を行う。

(D-7)補正された測位結果を返答する。

(D-8)測位コントローラは、補正された測位結果をリクエストコントローラに返答する。

(D-9)リクエストコントローラは、補正された測位結果をアプリケーションに返答する。

(E-1)指定された間隔が経過した後、リクエストコントローラは測位コントローラに再び現在位置取得を要求する。

(E-2)測位コントローラは、端末の状態やユーザの設定を鑑みて、引き続き測位手段 B を利用することを決める。

(E-3)測位手段 C のコントローラに現在位置取得要求を行う。

(E-4)測位手段コントローラは、測位コントローラに測位結果を返答する。

(E-5)測位コントローラは、リクエストコントローラに測位結果を返答する。

(E-6)リクエストコントローラは、アプリケーションに測位結果を返答する。

位置情報履歴に、足りないデータを補間するためにスムージング処理等を行い、軌跡として提供することができる。次に、その処理の流れを説明する。

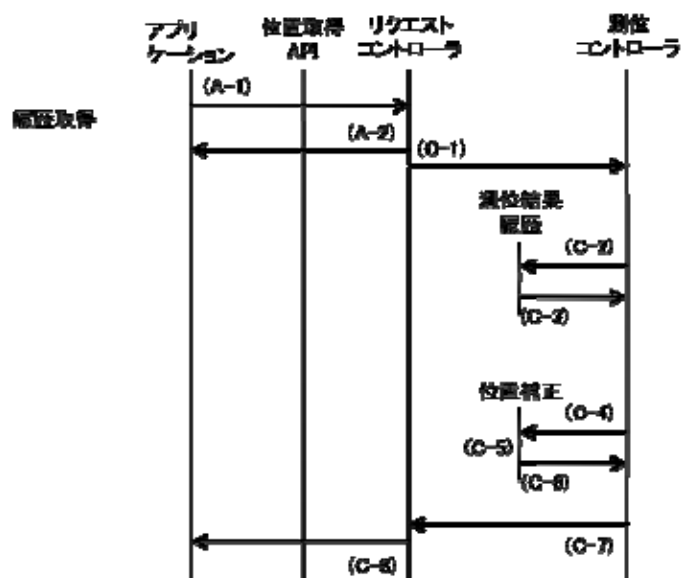


図 16：軌跡取得処理の流れ

(A-1)位置履歴取得処理を開始する。アプリケーションはリクエストコントローラに位置履歴取得を要求する。

(A-2)リクエストコントローラは、要求を正常に受けた旨を、即座に返答する。

(C-1)リクエストコントローラは、指定された期間の軌跡を測位コントローラに要求する。

(C-2)測位コントローラは、測位結果履歴モジュールに問い合わせる。

(C-3)測位結果履歴モジュールは、該当する測位結果の履歴を返答する。

(C-4)測位コントローラは、位置補正モジュールに、軌跡生成処理を要求する。

(C-5)例えば、測位結果履歴モジュールは、測位結果履歴が指定された間隔で存在しない場合に内挿処理を行う。または、外れ値判定なども行う。

(C-6)スムージング処理によって得られた軌跡を、測位コントローラに返答する。

(C-7)軌跡をリクエストコントローラに返答する。

(C-8)軌跡をアプリケーションに返答する。

測位手段モジュール内部の位置補正機能の共通処理基盤への組み込み方は実装依存とするが、次に挙げるモデルが考えられる。ある測位手段が、補正手段を2つもっているとして：

- 測位手段としてはひとつで、測位手段の設定機能により、利用する補正手段を選択する
- 補正手段毎に別の測位手段として共通処理基盤に登録する。この場合は、補正なし、補正手段1適用、補正手段2適用の3つの測位手段が存在することになる。

3.7. 適応的測位手段選択

共通処理基盤は、現在位置取得要求において明示的に測位手段を指定されなかった場合、端末の状態やユーザのポリシーを鑑みて、利用する測位手段を選択する。

共通処理基盤では、測位手段を適応的に判断する材料として、測位手段プロファイルおよび測位モード、要求精度、測位手段の状態などが利用可能である。測位手段選択アルゴリズムの例を、図 17 に示す。

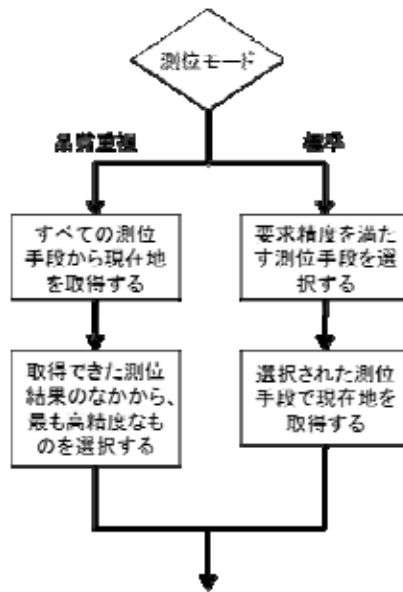


図 17：適応的測位手段アルゴリズム(参考)

なお、具体的な選択アルゴリズムは競争領域であるため本仕様では規定しない。

3.8. 測位結果履歴管理および軌跡生成

測位結果を測位手段コントローラが取得した際に、その測位結果は測位結果履歴モジュールに送られ、加工されずに記憶されるものとする。共通処理基盤が軌跡管理のために、定期的に現在位置取得をすることも考えられるが、本仕様ではそれを規定しない。

記憶された測位結果は、未加工のままアプリケーションが参照することも可能とする。また、内挿などの演算により等時間間隔の位置情報集合へと加工された軌跡としてアプリケーションが参照することも可能とする。

3.9. 測位手段抽象化

共通処理基盤に測位手段を登録する際には、測位手段コントローラを実装する。測位手段コントローラは、以下の機能を必ず含む。

- 第 6 章で規定するインターフェース
- 測位手段プロファイルの提供
- 測位手段状態の提供

共通処理基盤では、複数の測位手段を組み合わせ、別の測位手段としてみせるような測位手段コントローラの実装も許容する。

3.10. センサ情報の取り扱い

一般に、測位機能とセンサは緊密な関係にあり、各種のセンサは測位手段を構成する要素となり得る。例えば、加速度情報や地磁気情報等を活用することで、測位手段（自律測位と呼ばれる場合がある）を構成する事例がある。

このような構成の場合、地磁気および加速度の計測値は、センサ API に則りセンサミドルウェアから、測位手段へ直接提供されるものとする。したがって、共通処理基盤においては、位置情報を取得する機能に関連する API を提供するが、個々のセンサ情報を取得する API は含まれない。

単一のセンサが、複数の測位手段によって共有されるケースについては、センサミドルウェアがリソース管理を行うものとし、共通処理基盤の機能に含まない。

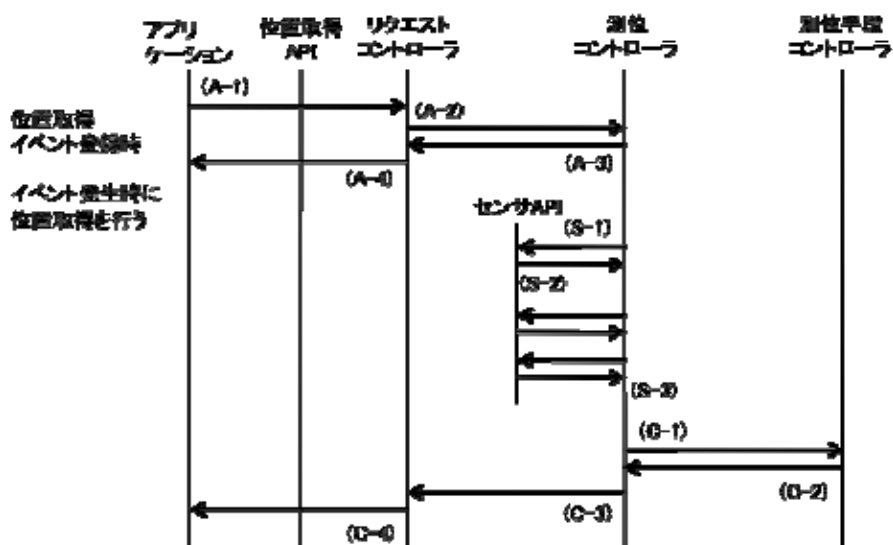


図 18： センサ情報を利用したイベント駆動位置取得処理の流れ

(A-1)非同期 API で位置取得イベントを登録することにより、イベント駆動による現在位置取得を要求する。

(A-2)リクエストコントローラは、測位コントローラに位置取得イベントを登録を要求する。

(A-3)正常に要求を受けた場合は、即座に返答する。

(A-4)正常に要求を受けた旨を、アプリケーションに返答する。

-
- (S-1)測位コントローラは、定期的にセンサ計測結果を要求する。
 - (S-2)センサ計測結果を測位コントローラに返答する。
 - (S-3)測位コントローラは、センサ計測結果を登録された位置取得イベントの条件式により評価する。

(C-1)位置取得イベントの条件評価結果が真の場合、測位コントローラは、登録されている当該位置取得イベントを削除し、測位手段コントローラに現在位置取得を要求する。

(C-2)測位手段コントローラは、測位手段に現在位置取得を要求し、測位結果の返答を受け、測位結果を測位コントローラに返答する。

(C-3)測位結果をリクエストコントローラに返答する。

(C-4)測位結果をアプリケーションにコールバック返答する。

3.11. 測位結果へのフィードバック

現在位置取得処理の結果が間違っていると判断される場合、その旨をアプリケーションから共通処理基盤に、もしくは共通処理基盤から測位手段にフィードバックできると、位置情報の詐称に対する防御や測位精度向上に役立てることができる。しかし、具体的なインタフェースは競争領域とし、本仕様書では定義しない。

4. データ構造

4.1. 位置表現

共通処理基盤では、様々な位置表現を取り扱うため、位置表現を空間識別子と位置情報の対で管理する。また、位置表現変換モジュール内では、各位置表現間のマッピングルールを保持し、必要に応じて位置表現を変換して利用する。図 19 はその概念図である。



図 19: 位置表現概念

共通処理基盤の位置表現は、基本的に場所識別子 PI (Place Identifier) [2]の仕様と合致している。位置表現変換は、PI においては、PI 変換に相当する。また、位置表現変換を行うため、空間識別子を設けている。この空間は、PI の空間参照系に基づく。マッピングルールは、PI における PI_ConversionServiceRule に相当する。したがって、共通処理基盤の、位置表現変換モジュールは、PI 基盤と連携、または PI 基盤そのものへのインタフェースと捉えることができる。

通常、アプリケーションは PI サービスを利用するため PI 基盤に問い合わせを行う。PI 基盤にはレジストリなどが用意されており、適切な PI サービスを探す手段が用意されている。PI の枠組みでは、アプリケーションはこのような PI サービスを通じて、PI の変換を行うことができる。共通処理基盤においては、位置表現変換モジュールが、必要に応じて PI 基盤と連携する役割を果たす。

アプリケーションはさまざまな要求をもつため、必要とする位置表現も異なる。緯度経度と現在位置を道路地図にプロットするようなアプリケーションから、地図が整備されていない屋内エリア/建物内で位置情報を文字列/音声で表現してナビゲーションするようなものも考えられる。共通処理基盤がサポートする測位手段が提供できる位置表現も複数あり、アプリケーションの要求をそのまま満たすことができれば、変換せずその情報をアプリケーションに提供する。しかし、測位手段が提供できる位置表現も限られているので、マッピングルール(例えば測地系の変換ルーチン、座標変換計算式や、緯度経度から住所を得る

ための変換 ASP)や PI 基盤などとの連携は不可欠となる。

最終的に、共通処理基盤では、アプリケーションの位置表現要求(緯度経度、局所座標系、座標系とは全く異なる任意の文字列表現など)を満たすために、変換が必要ない場合は、適切な測位手段を選択し、アプリケーションへその位置情報を提供するし、適切な測位手段がない場合は、位置表現変換モジュールを通じて変換が行われる。

4.2. 測位手段プロファイル

共通処理基盤では、様々な測位手段を取り扱うために、各測位手段特性を示す測位手段プロファイルにより管理する。測位手段プロファイルは、測位手段コントローラモジュールにより管理され、他のモジュールに提供される。アプリケーションは、測位手段プロファイルを参照することで、明示的に測位方式を選択することも可能である。

測位手段プロファイルは、次に挙げる情報を含むこと。

- 測位手段識別子
- 測位手段名
 - GPS、屋内 GPS、PlaceEngine 等、測位手段を表す名称。
- 測位精度
 - 測位手段が提供する位置情報が含む最大誤差等、測位結果の空間的な正確さを表す指標。水平成分と高さ成分により構成される。
- 利用可能な位置表現
 - 測位手段が提供する位置情報を表現するに当たり、利用可能な表現方法。
 - 例えば、一般的な GPS モジュールが提供する位置情報は、経緯度および標高を世界測地系における座標値としてされ得るが、通常は地名として表現されない。(この例は必要?)
- 利用可能な位置表現における空間分解能
 - 利用可能な個々の位置表現方法に対応した、表現の最小単位。
 - 例えば、経緯度について「0.000001 度」「1 ミリ秒」等。高度について「メートル」等。また、地名について「番地」「フロア名」等。
- 測位結果更新間隔 (測位の連続性)
 - 測位手段が定期的に測位を継続する際の、測位結果が更新される標準的な時間的間隔
- 測位時刻分解能
 - 測位手段が提供する測位時刻 (位置情報と対となる時刻) の最小単位。

-
- 例えば、「0.1 秒」等。
 - 測位結果信頼度
 - 測位手段が提供する位置情報が含む最大誤差や確度等を総合的に表す指標。
 - 運用性
 - 消費電力
 - ◇ 測位手段が消費する標準的な電力
 - 利用環境/条件
 - 屋内・屋外
 - ◇ 測位手段が屋内あるいは屋外で利用が推奨されるか否かの別。
 - 通信の必要性
 - ◇ 測位手段が外部との通信を必要とするか否かの別。
 - アクセス権限
 - ◇ 測位手段の利用が許可され得る資格情報

4.3. 測位結果

測位結果は、以下の情報を含むものとする。

- 位置表現
 - 詳しくは、第 4.1 節参照
- 測位時刻
 - 定義は、第 2 章を参照
- 測位精度
 - 定義は、第 2 章を参照
- 測位手段識別子または測位手段名
 - 共通処理基盤内で測位手段を一意に識別するための情報
- 実際に適用された測位モード
 - 測位モードの定義は、第 4.5 節を参照
- 測位に要した時間
- きっかけとなったイベント
 - イベントの定義は、第 4.6 節を参照

この情報は一つの点を表すもので、測位結果履歴を表すときはこれのリストになる。

4.4. 軌跡

軌跡は、以下の情報を含むものとする。

- 軌跡要素の集合
 - 記憶されている測位結果履歴を内挿等により加工することで新たに算出された点列（位置情報の列）を表す。
- 空間識別子
 - 軌跡を構成する各軌跡要素の位置情報を表現する空間の識別子。

軌跡要素は、以下の情報を含むものとする。

- 位置情報
 - 点列を構成するひとつひとつの点などの場所。
- 時刻
 - ひとつひとつの位置情報に対応する時刻。

4.5. 測位モード

共通処理基盤では、測位手段を選択する際のユーザのポリシーを、測位モードとして抽象化する。どのような測位モードを設定するかは競争領域であり、本仕様では定義しない。ただし、本仕様書内では、便宜上、以下の二つのモードを設定する。

- 品質重視モード
 - 現在位置取得機能呼び出した際に、できるだけ精度の高い測位結果を取得しようとするモード
- 通常測位モード
 - 上記以外のポリシーで、測位手段を選択するモード

4.6. イベント

共通処理基盤では、次に挙げるイベントを取り扱う。

- 入退域判定イベント
 - ある空間参照系のある地域に入ったこと、あるいは出たことを示すイベント
- センサイベント
 - センサから得られる情報がある条件を満たしたことを示すイベント
 - センサ情報と判定式の組み合わせで表現する。

-
- 測位手段状態変化イベント
 - 共通処理基盤が管理する測位手段の状態が変更されたことを示すイベント
 - プロファイルの変更やデバイスの取り外しが上記に該当する。

4.7. 測位手段状態

共通処理基盤で扱う測位手段の状態は、次のとおりである。

- READY
 - 測位手段が利用可能である状態
- NOT_AVAILABLE
 - 測位手段が利用不可である状態
- INITIALIZING
 - 測位手段が利用可能であるが、測位手段が起動/初期化処理中である状態
- ERROR
 - 測位手段がエラーを起こしている状態

4.8. バージョン

共通処理基盤仕様のバージョンを規定する。バージョンは、メジャーバージョンとマイナーバージョンからなるものとする。なお、本仕様はバージョン 1.0 である。

5. アプリケーションインタフェース

本章では、共通処理基盤とアプリケーション間のインタフェースを規定する。メソッド名は例であり、該当する機能を遂行するためのメソッドを規定するものではない。

インタフェースは次の凡例のように記述する。

- a) メソッド名
- b) 機能概要
- c) パラメタ
- d) 戻り値
- e) 補足説明

なお、全てのメソッドについて、d)戻り値 に記述した値に加え、エラー識別子が返される。

エラー識別子の具体的な定義は本書では割愛するが、第 3.2.7 節で述べたアクセスコントロール判定の可否については、必ず含めること。

5.1. 位置取得 API

a) `getCurrentLocation`

- b) 現在位置を取得する。
- c) 測位手段識別子(のリストと優先度)、位置補正のありなし、タイムアウト時間、空間識別子
- d) 測位結果(のリスト)
- e) 現在位置取得の blocking I/O である。パラメタが指定されない場合は、共通処理基盤が適切と判断した測位手段の結果を返す。共通処理基盤で補正する前の測位手段から得られた測位結果そのものを取得したい場合は、位置補正なしを指定する。

a) `getCurrentLocationAsynchronous`

- b) 現在位置を取得する。
- c) コールバック先インスタンス、測位手段識別子(のリストと優先度)、位置補正のありなし、タイムアウト時間、空間識別子
- d) 測位結果(のリスト)
- e) 現在位置取得の non-blocking I/O である。パラメタが指定されない場合は、共通処理基盤が適切と判断した測位手段の結果を返す。共通処理基盤で補正する前の測位手段から得られた測位結果そのものを取得したい場合は、位置補正なしを指定する。

a) `startUpdatingLocation`

- b) 定期的に、あるいはイベント発生時に現在位置を取得する。
- c) コールバック先インスタンス、測位手段識別子(のリストと優先度)、位置補正のありなし、空間識別子
- d) なし
- e) インターバルやイベントの登録には別の API を用いる。

a) `stopUpdatingLocation`

- b) 定期的な、あるいはイベント駆動な現在位置取得を終了する。
- c) なし
- d) なし
- e) 特になし。

-
- a) getLocationHistory
 - b) 測位結果履歴を取得する。
 - c) 取得対象の開始時刻、取得対象の終了時刻
 - d) 測位結果の集合
 - e) 特になし。

- a) deleteLocationHistory
- b) 記憶されている測位結果履歴を完全に削除する。
- c) なし
- d) なし
- e) 特になし。

- a) getTrajectory
- b) 軌跡を取得する。
- c) 取得対象の開始時刻、取得対象の終了時刻、取得対象の時間的間隔、軌跡を表現する空間識別子、軌跡生成アルゴリズム識別子
- d) 軌跡
- e) 軌跡生成アルゴリズム識別子が省略された場合には、setDesiredTrajectoryInterpolationMethodにより指定された軌跡生成アルゴリズムが使用される。(取得対象の開始・終了時刻が測位結果履歴の記憶対象期間を外れる場合は外挿とする。)

5.2. 位置表現変換 API

- a) convertLocation
- b) 位置表現を変換する。
- c) 変換前の位置表現、変換後の空間識別子
- d) 変換後の位置表現
- e) 位置表現変換の blocking I/O である。

- a) convertLocationAsynchronous
- b) 位置表現を変換する。
- c) 変換前の位置表現、変換後の空間識別子
- d) 変換後の位置表現
- e) 位置表現変換の non-blocking I/O である。

5.3. コントロール API

a) `getAllPositioningMethods`

b) 利用可能な測位手段のリストを取得する。

c) なし

d) 測位手段識別子と測位手段の状態のリスト

e) 特になし。

a) `getPositioningMethodProfile`

b) 測位手段のプロファイルを取得する。

c) 測位手段識別子

d) 測位手段のプロファイル

e) 特になし。

a) `setDesiredPositioningMode`

b) 測位モードを指定する。

c) 測位モード識別子(のリスト)

d) なし

e) 測位手段を指定しないで現在位置を取得した場合のポリシーを指定する。

a) `getAllPositioningModes`

b) 利用可能な測位モードのリストを取得する。

c) なし

d) 測位モード識別子のリスト

e) 特になし。

a) `getAllLocationFormats`

b) 扱える位置表現のリストを取得する。

c) なし

d) 空間識別子のリスト

e) 特になし。

a) `setDesiredLocationFormat`

b) 主に扱いたい位置表現を指定する。

c) 空間識別子

d) なし

e) 特なし。

a) setPositioningInterval

b) 定期的に位置情報を取得する際のインターバルを指定する。

c) なし

d) なし

e) 特になし。

a) getPositioningInterval

b) 現在位置取得コールバック間のインターバルを取得する。

c) なし

d) インターバル

e) 特になし。

a) setHistoryLength

b) 測位結果履歴を記憶する時間的長さを指定する。

c) 時間的長さ

d) なし

e) 特になし。

a) getHistoryLength

b) 測位結果履歴を記憶する時間的長さを取得する。

c) なし

d) 時間的長さ

e) 特になし。

a) getAllTrajectoryInterpolationMethods

b) 軌跡内の点を補間するためのアルゴリズムのリストを取得する。

c) なし

d) 補間アルゴリズム識別子のリスト

e) 特になし。

a) setDesiredTrajectoryInterpolationMethod

b) 軌跡内の点を補間するためのデフォルトのアルゴリズムを指定する。

c) 補間アルゴリズム識別子

d) なし

e) 特になし。

a) getAllEvents

b) イベント駆動で位置を取得する際のシステムに登録しているすべてのイベントを取得する。

c) なし

d) イベント識別子とイベント記述のリスト

e) 特になし。

a) setEvent

b) イベントを登録する。

c) イベント記述

d) イベント識別子

e) 特になし。

a) deleteEvent

b) イベントを登録解除する。

c) イベント識別子

d) なし

e) 特になし。

a) getVersion

b) 適合する共通処理基盤仕様のバージョン情報を取得する。

c) なし

d) バージョン識別子

e) 特になし。

6. 測位手段コントローラインタフェース

本章では、測位手段コントローラと測位コントローラ間のインタフェースを規定する。凡例は、第5章と同等である。

a) getCurrentLocation

b) 現在位置を取得する。

c) タイムアウト時間

d) 測位結果(のリスト)

e) 現在位置取得の blocking I/O である。

a) startUpdatingLocation

b) 非同期に現在位置を取得する。

c) コールバック先インスタンス

d) なし

e) 特になし。

a) stopUpdatingLocation

b) 非同期な現在位置取得を終了する。

c) なし

d) なし

e) 特になし。

a) getState

b) 測位手段の状態を取得する。

c) なし

d) 状態識別子

e) 特になし。

参考文献

- [1] “ GPS信号を応用した屋内外シームレス位置情報システムの開発 ”, 川口 貴正, 江端 智一, 吉原 慎哉, 下垣 豊, 情報処理学会 DICOMO2008 Proceedings
- [2] 財団法人日本情報処理開発協会, PI仕様書, 2008.3

変更履歴

版数	日付	改訂内容	担当者
0.1	2009/12/17	初版作成	塩野崎, 荒牧
0.2	2010/1/12	3, 4.1, 4.2.4, 4.3.4, 4.3.5, 4.6 を拡充	塩野崎, 荒牧
0.3	2010/1/19	元 3 章を 1 章にマージ。 3.2.9, 3.2.10, 3.2.11 を追記 6 章を追記	塩野崎
1.0	2010/2/26	3.11 を追記 5,6 のインターバル設定に関する API を見直し 5.3 に基盤バージョン取得 API を追加	塩野崎、荒牧