

位置管理モジュールおよび環境サーバ実装のためのシステム設計

○河 寅勇(東京大学) 田村 雄介(東京大学) 森下 壮一郎(東京大学) 浅間 一(東京大学)

岡本 浩幸(ライテックス) 野田 五十樹(産業技術総合研究所) 羽田 靖史(情報通信研究機構)

System Design for Implementation of Location Management Module and Environment Server

*Inyong HA^{*1}, Yusuke TAMURA^{*1}, Soichiro MORISHITA^{*1}, Hajime ASAMA^{*1} ^{*1}The University of Tokyo

Hiroyuki OKAMOTO(Ritecs, Inc.), Itsuki NODA(AIST) and Yasushi HADA(NICT)

Abstract — In this paper, we propose object information service system which can provide the information of object such as location, speed, measurement error and etc. For the implementation of the proposed system, we designed the Location Management Module for registering information and the Environment Server for providing. In addition, communication protocol and API for accessing each module were also defined. To verify the system, we tested designed module and API on the assumption that several moving objects existed in the same environment.

Key Words: Intelligent Environment Space, Location Management Module, Environment Server

1. 序 論

ロボットシステムが環境変化に対して頑健にタスクを遂行するためには環境を認識し情報を共有できる知能化環境が必要である[1]。この知能化環境構築の一環として我々は環境情報の一つであるオブジェクトの位置や速度等を管理・提供する位置管理モジュールと環境サーバの設計を行った[2]。本稿では提案したシステムを実装して運用するために、その実装方法を検討し、情報登録および利用プロトコルに基づいた外部提供用 API の設計を行う。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通して設計したシステムと API の動作を検証する。

2. システムの構成

位置管理モジュールおよび環境サーバを含む全体システムの構成例を図 1 に示す。

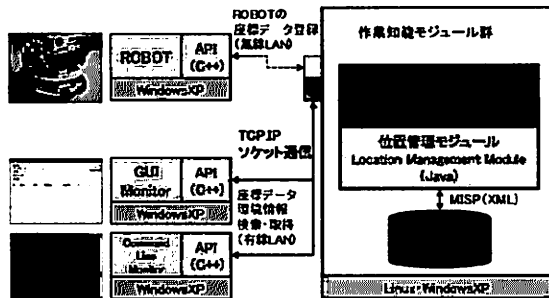


図 1 全体システムの構成例

2.1 全体システム構成

環境内に存在するロボットやセンサ等のユーザは、提供される API を利用してオブジェクトの情報を登

録および検索する。情報検索の応用ソフトウェアも同一 API を利用している。API と位置管理モジュール・環境サーバとの間は TCP ソケットで、位置管理モジュール・環境サーバとデータベースである DaRuMa[3]との間は DaRuMa/MISP[4]でそれぞれ通信を行う。

2.2 位置管理モジュール・環境サーバの実装

位置管理モジュールは、ユーザからのオブジェクト位置情報の登録、環境サーバは位置情報の検索および取得に関するリクエストに対応する[2]。このモジュール群は Java で実装されており、Linux および Windows の上で動作可能である。環境サーバの役割として単純な情報提供だけではなく、より上位レベルの提供方法、つまりイベント方式やバルクデータ転送方式などを現段階で検討している。

2.3 TMS との連携方法の提案

システムの拡張性と互換性を高めるために他の位置管理サービスシステムである Town Management System (TMS) [5]との連携方法を提案する。相互に提供されている API をシステム中で変換するモジュールを搭載する概念を図 2 に示した。現在基本的な機能の相互利用に関する議論が進んでいる。

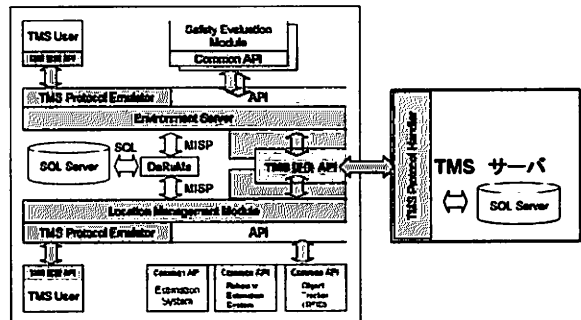


図 2 TMS との連携構造

2.4 座標変換の対応

あらかじめ座標系間の変換行列を登録しておくことで、ローカル座標系で登録されたオブジェクトでも任意の座標系における位置情報を得られるようにした。変換行列は図3のような二次元の回転と並進を表せるように2×3行列として定義した。

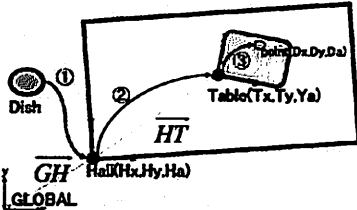


図3 変換行列による座標の取得の例

3. API の設計

位置管理モジュール・環境サーバとの通信はそのため提供されるAPIで行われる。APIの設計のためにプロトコルの形式と手順を定義した。

3.1 通信プロトコル

座標系の登録からオブジェクトの情報検索までのプロトコルの形式と手順を図4に示す。通信プロトコルの形式には、TCP/IPのストリームデータの内容としてオブジェクトのIDと速度、測定誤差などが含まれている。タイムスタンプも記述することで、移動オブジェクトの管理にも対応できるようにした。

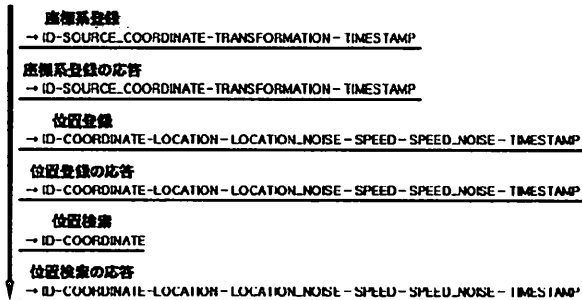


図4 通信プロトコルの形式および手順

3.2 API 関数の設計

以上で定義した通信プロトコルに基づいてAPI関数を設計しC++で実装した。拡張性を高めるためにJavaへの移植を進めている。座標登録、位置登録、位置検索に該当する代表的なAPI関数を図5に示す。APIは登録や検索などの様々なリクエストをTCP/IPのストリームデータに変換して位置管理モジュール・環境サーバと通信を行う。

名称	機能	書式
座標系登録	座標系を登録する	ip.coord.registerCoordinateSystem()
位置登録	位置を登録する	ip.coord.registerObjectLocation()
位置検索	オブジェクトの最新位置を取得する (最新時間の位置)	ip.coord.getLatestObjectLocation()
	オブジェクトの位置履歴を取得する (最新位置から順数を指定して取得)	

図5 主なAPI関数の例

4. 実験

複数の移動ロボットが同じ環境に存在することを仮定し、オブジェクト登録および取得を実際に行って実装したシステムの動作を検証した。図6はその結果として移動オブジェクトの軌跡が管理されている様子を示している。視覚的に軌跡を観察するためにDaRuMa Earth[4]を利用した。

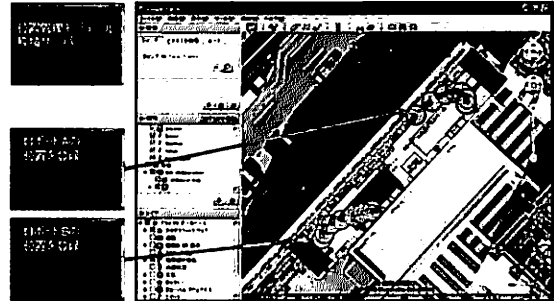


図6 複数移動オブジェクトの位置管理の例

5. 結論

本稿では提案した位置管理モジュール・環境サーバを実装するため、その実装方法とオブジェクトの情報サービスAPIの設計について検討した。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通してシステムの動作を検証した。

今後の展望として、実装されたモジュールをRTC化[6]、他のモジュール群との連動テストなどを予定している。

謝辞 本研究開発はH20年度NEDO次世代ロボット知能化技術開発プロジェクトの一部として実施されたものである。

- [1] 松日榮 他: “ロボタストに作業を実行するための作業知能モジュール群の開発: プロジェクト概要と進捗”, 第26回日本ロボット学会学術講演会, 展開セッション9, 2008.
- [2] 河 他: “知能化環境構築のための位置管理モジュール及び環境サーバの設計”, 第26回日本ロボット学会学術講演会, RSJ2008AC1F2-08, 2008.
- [3] 野田 他: “減災情報共有データベースによる災害情報システム統合”, 第21回人工知能学会全国大会予稿集, 人工知能学会, 宮崎, pp. 2E5-6 1-4, 2007.
- [4] 下羅 他: “災害時情報共有のためのプロトコルおよびその実装”, 人工知能学会 社会におけるAI研究会(第四回), pp. SIG-SAI-004-02 1-7, 2007.
- [5] 馬場 他: “ロボットタウンの実証的研究—タウンマネジメントシステムの開発—”, 第24回日本ロボット学会学術講演会予稿集 2N15, 9, 2006
- [6] OMG (Object Management Group): “The Robotic Technology Component Specification,” http://www.omg.org/technology/documents/domain_spec_catalog.htm

位置管理モジュールおよび環境サーバ実装のためのシステム設計

○河 寅勇(東京大学) 田村 雄介(東京大学) 森下 壮一郎(東京大学) 浅間 一(東京大学)

岡本 浩幸(ライテックス) 野田 五十樹(産業技術総合研究所) 羽田 靖史(情報通信研究機構)

System Design for Implementation of Location Management Module and Environment Server

*Inyong HA^{*1}, Yusuke TAMURA^{*1}, Soichiro MORISHITA^{*1}, Hajime ASAMA^{*1} ^{*1}The University of Tokyo

Hiroyuki OKAMOTO(Ritecs, Inc.), Itsuki NODA(AIST) and Yasushi HADA(NICT)

Abstract — In this paper, we propose object information service system which can provide the information of object such as location, speed, measurement error and etc. For the implementation of the proposed system, we designed the Location Management Module for registering information and the Environment Server for providing. In addition, communication protocol and API for accessing each module were also defined. To verify the system, we tested designed module and API on the assumption that several moving objects existed in the same environment.

Key Words: Intelligent Environment Space, Location Management Module, Environment Server

1. 序論

ロボットシステムが環境変化に対して頑健にタスクを遂行するためには環境を認識し情報を共有できる知能化環境が必要である[1]。この知能化環境構築の一環として我々は環境情報の一つであるオブジェクトの位置や速度等を管理・提供する位置管理モジュールと環境サーバの設計を行った[2]。本稿では提案したシステムを実装して運用するために、その実装方法を検討し、情報登録および利用プロトコルに基づいた外部提供用 API の設計を行う。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通して設計したシステムと API の動作を検証する。

2. システムの構成

位置管理モジュールおよび環境サーバを含む全体システムの構成例を図1に示す。

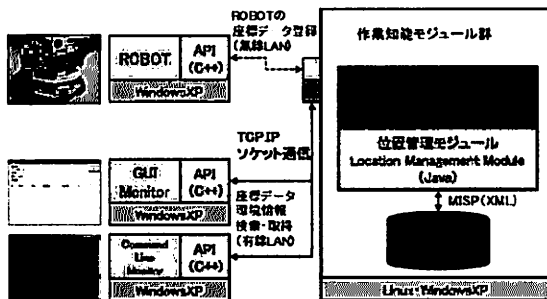


図1 全体システムの構成例

2.1 全体システム構成

環境内に存在するロボットやセンサ等のユーザは、提供される API を利用してオブジェクトの情報を登

録および検索する。情報検索の応用ソフトウェアも同一 API を利用している。API と位置管理モジュール・環境サーバの間は TCP ソケットで、位置管理モジュール・環境サーバとデータベースである DaRuMa[3]との間は DaRuMa/MISP[4]でそれぞれ通信を行う。

2.2 位置管理モジュール・環境サーバの実装

位置管理モジュールは、ユーザからのオブジェクト位置情報の登録、環境サーバは位置情報の検索および取得に関するリクエストに対応する[2]。このモジュール群は Java で実装されており、Linux および Windows の上で動作可能である。環境サーバの役割として単純な情報提供だけではなく、より上位レベルの提供方法、つまりイベント方式やバルクデータ転送方式などを現段階で検討している。

2.3 TMS との連携方法の提案

システムの拡張性と互換性を高めるために他の位置管理サービスシステムである Town Management System (TMS) [5]との連携方法を提案する。相互に提供されている API をシステム中で変換するモジュールを搭載する概念を図2に示した。現在基本的な機能の相互利用に関する議論が進んでいる。

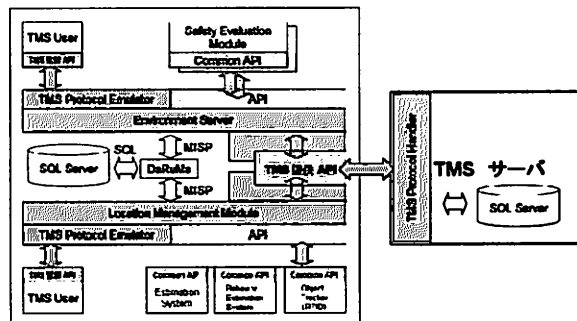


図2 TMS との連携構造

2.4 座標変換の対応

あらかじめ座標系間の変換行列を登録しておくことで、ローカル座標系で登録されたオブジェクトでも任意の座標系における位置情報を得られるようにした。変換行列は図3のような二次元の回転と並進を表せるように2×3行列として定義した。

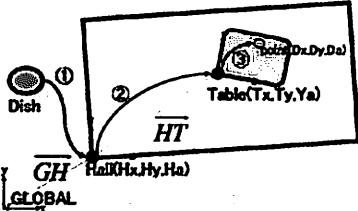


図3 変換行列による座標の取得の例

3. API の設計

位置管理モジュール・環境サーバとの通信はそのために提供されるAPIで行われる。APIの設計のためにプロトコルの形式と手順を定義した。

3.1 通信プロトコル

座標系の登録からオブジェクトの情報検索までのプロトコルの形式と手順を図4に示す。通信プロトコルの形式には、TCP/IPのストリームデータの内容としてオブジェクトのIDと速度、測定誤差などが含まれている。タイムスタンプも記述することで、移動オブジェクトの管理にも対応できるようにした。

座標系登録	→ ID-SOURCE_COORDINATE-TRANSFORMATION-TIMESTAMP
座標系登録の応答	→ ID-SOURCE_COORDINATE-TRANSFORMATION-TIMESTAMP
位置登録	→ ID-COORDINATE-LOCATION-LOCATION_NOISE-SPEED-SPEED_NOISE-TIMESTAMP
位置登録の応答	→ ID-COORDINATE-LOCATION-LOCATION_NOISE-SPEED-SPEED_NOISE-TIMESTAMP
位置検索	→ ID-COORDINATE
位置検索の応答	→ ID-COORDINATE-LOCATION-LOCATION_NOISE-SPEED-SPEED_NOISE-TIMESTAMP

図4 通信プロトコルの形式および手順

3.2 API 関数の設計

以上で定義した通信プロトコルに基づいてAPI関数を設計しC++で実装した。拡張性を高めるためにJavaへの移植を進めている。座標登録、位置登録、位置検索に該当する代表的なAPI関数を図5に示す。APIは登録や検索などの様々なリクエストをTCP/IPのストリームデータに変換して位置管理モジュール・環境サーバと通信を行う。

名称	機能	書式
座標系登録	座標系を登録する	ip.oc.rtcasmp.registerCoordinateSystem0
位置登録	位置を登録する	ip.oc.rtcasmp.setObjectLocation0
位置検索	オブジェクトの最新位置を取得する (最新時刻の位置)	ip.oc.rtcasmp.getObjLocatior0
	オブジェクトの位置履歴を取得する (最新位置から順数を指定して取得)	

図5 主なAPI関数の例

4. 実験

複数の移動ロボットが同じ環境に存在することを仮定し、オブジェクト登録および取得を実際に行って実装したシステムの動作を検証した。図6はその結果として移動オブジェクトの軌跡が管理されている様子を示している。視覚的に軌跡を観察するために DaRuMa Earth[4]を利用した。

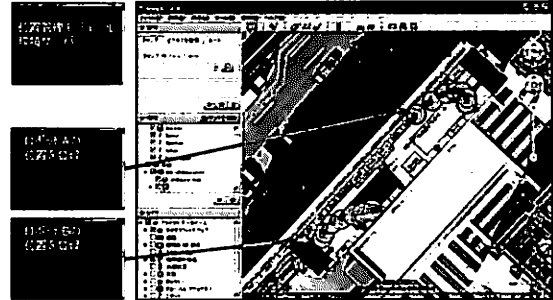


図6 複数移動オブジェクトの位置管理の例

5. 結論

本稿では提案した位置管理モジュール・環境サーバを実装するため、その実装方法とオブジェクトの情報サービスAPIの設計について検討した。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通してシステムの動作を検証した。

今後の展望として、実装されたモジュールをRTC化[6]、他のモジュール群との連動テストなどを予定している。

謝辞 本研究開発はH20年度NEDO次世代ロボット知能化技術開発プロジェクトの一部として実施されたものである。

- [1] 松日榮 他: “ロボスタに作業を実行するための作業知能モジュール群の開発: プロジェクト概要と進捗”, 第26回日本ロボット学会学術講演会, 展開セッション9, 2008.
- [2] 河 他: “知能化環境構築のための位置管理モジュール及び環境サーバの設計”, 第26回日本ロボット学会学術講演会, RSJ2008AC1F2-08, 2008.
- [3] 野田 他: “減災情報共有データベースによる災害情報システム統合”, 第21回人工知能学会全国大会予稿集, 人工知能学会, 宮崎, pp. 2E5-6 1-4, 2007.
- [4] 下羅 他: “災害時情報共有のためのプロトコルおよびその実装”, 人工知能学会 社会におけるAI研究会(第四回), pp. SIG-SAI-004-02 1-7, 2007.
- [5] 馬場 他: “ロボットタウンの実証的研究—タウンマネジメントシステムの開発—”, 第24回日本ロボット学会学術講演会予稿集 2N15, 9, 2006
- [6] OMG (Object Management Group): “The Robotic Technology Component Specification,” http://www.omg.org/technology/documents/domain_spec_catalog.htm

位置管理モジュールおよび環境サーバ実装のためのシステム設計

○河 寅勇(東京大学) 田村 雄介(東京大学) 森下 壮一郎(東京大学) 浅間 一(東京大学)
 岡本 浩幸(ライテックス) 野田 五十樹(産業技術総合研究所) 羽田 靖史(情報通信研究機構)

System Design for Implementation of Location Management Module and Environment Server

*Inyong HA^{*1}, Yusuke TAMURA^{*1}, Soichiro MORISHITA^{*1}, Hajime ASAMA^{*1} ^{*1}The University of Tokyo

Hiroyuki OKAMOTO(Ritecs, Inc.), Itsuki NODA(AIST) and Yasushi HADA(NICT)

Abstract — In this paper, we propose object information service system which can provide the information of object such as location, speed, measurement error and etc. For the implementation of the proposed system, we designed the Location Management Module for registering information and the Environment Server for providing. In addition, communication protocol and API for accessing each module were also defined. To verify the system, we tested designed module and API on the assumption that several moving objects existed in the same environment.

Key Words: Intelligent Environment Space, Location Management Module, Environment Server

1. 序論

ロボットシステムが環境変化に対して頑健にタスクを遂行するためには環境を認識し情報を共有できる知能化環境が必要である[1]。この知能化環境構築の一環として我々は環境情報の一つであるオブジェクトの位置や速度等を管理・提供する位置管理モジュールと環境サーバの設計を行った[2]。本稿では提案したシステムを実装して運用するために、その実装方法を検討し、情報登録および利用プロトコルに基づいた外部提供用 API の設計を行う。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通して設計したシステムと API の動作を検証する。

2. システムの構成

位置管理モジュールおよび環境サーバを含む全体システムの構成例を図 1 に示す。

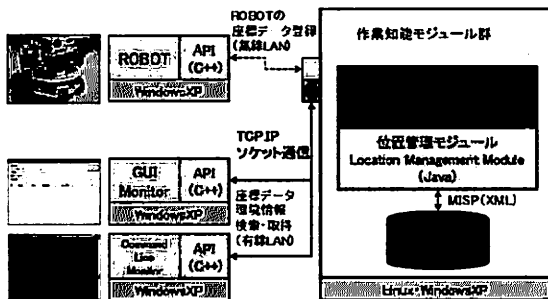


図 1 全体システムの構成例

2.1 全体システム構成

環境内に存在するロボットやセンサ等のユーザは、提供される API を利用してオブジェクトの情報を登

録および検索する。情報検索の応用ソフトウェアも同一 API を利用している。API と位置管理モジュール・環境サーバとの間は TCP ソケットで、位置管理モジュール・環境サーバとデータベースである DaRuMa[3]との間は DaRuMa/MISP[4]でそれぞれ通信を行う。

2.2 位置管理モジュール・環境サーバの実装

位置管理モジュールは、ユーザからのオブジェクト位置情報の登録、環境サーバは位置情報の検索および取得に関するリクエストに対応する[2]。このモジュール群は Java で実装されており、Linux および Windows の上で動作可能である。環境サーバの役割として単純な情報提供だけではなく、より上位レベルの提供方法、つまりイベント方式やバルクデータ転送方式などを現段階で検討している。

2.3 TMS との連携方法の提案

システムの拡張性と互換性を高めるために他の位置管理サービスシステムである Town Management System (TMS) [5]との連携方法を提案する。相互に提供されている API をシステム中で変換するモジュールを搭載する概念を図 2 に示した。現在基本的な機能の相互利用に関する議論が進んでいる。

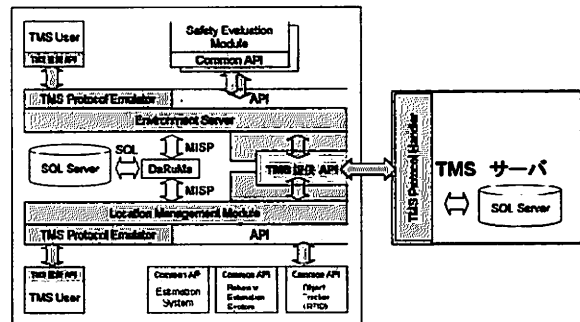


図 2 TMS との連携構造

2.4 座標変換の対応

あらかじめ座標系間の変換行列を登録しておくことで、ローカル座標系で登録されたオブジェクトでも任意の座標系における位置情報を得られるようにした。変換行列は図3のような二次元の回転と並進を表せるように2×3行列として定義した。

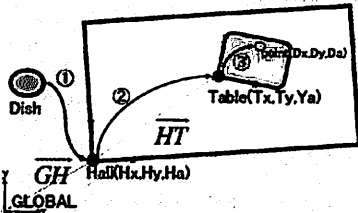


図3 変換行列による座標の取得の例

3. API の設計

位置管理モジュール・環境サーバとの通信はそのために提供されるAPIで行われる。APIの設計のためにプロトコルの形式と手順を定義した。

3.1 通信プロトコル

座標系の登録からオブジェクトの情報検索までのプロトコルの形式と手順を図4に示す。通信プロトコルの形式には、TCP/IPのストリームデータの内容としてオブジェクトのIDと速度、測定誤差などが含まれている。タイムスタンプも記述することで、移動オブジェクトの管理にも対応できるようにした。

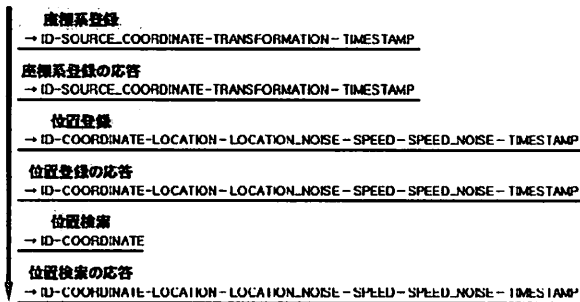


図4 通信プロトコルの形式および手順

3.2 API 関数の設計

以上で定義した通信プロトコルに基づいてAPI関数を設計しC++で実装した。拡張性を高めるためにJavaへの移植を進めている。座標登録、位置登録、位置検索に該当する代表的なAPI関数を図5に示す。APIは登録や検索などの様々なリクエストをTCP/IPのストリームデータに変換して位置管理モジュール・環境サーバと通信を行う。

名称	機能	書式
座標系登録	座標系を登録する	<code>ipcooritscm.registerCoordinateSystem()</code>
位置登録	位置を登録する	<code>ipcooritscm.registerObjectLocation()</code>
位置検索	オブジェクトの最新位置を取得する (最新時刻の位置) オブジェクトの位置履歴を取得する (最新位置から順数を指定して取得)	<code>ipcooritscm.getObjctLocation()</code>

図5 主なAPI関数の例

4. 実験

複数の移動ロボットが同じ環境に存在することを仮定し、オブジェクト登録および取得を実際に行って実装したシステムの動作を検証した。図6はその結果として移動オブジェクトの軌跡が管理されている様子を示している。視覚的に軌跡を観察するためにDaRuMa Earth[4]を利用した。

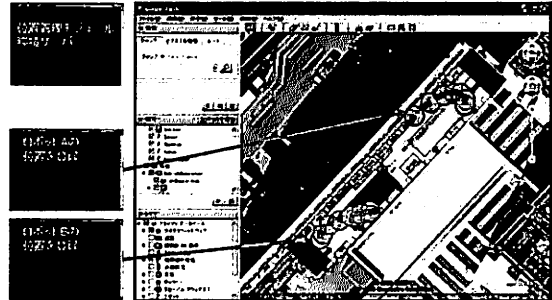


図6 複数移動オブジェクトの位置管理の例

5. 結論

本稿では提案した位置管理モジュール・環境サーバを実装するため、その実装方法とオブジェクトの情報サービスAPIの設計について検討した。さらに、複数の移動オブジェクトの位置情報サービスについての実験を通してシステムの動作を検証した。

今後の展望として、実装されたモジュールをRTC化[6]、他のモジュール群との連動テストなどを予定している。

謝辞 本研究開発はH20年度NEDO次世代ロボット知能化技術開発プロジェクトの一部として実施されたものである。

- [1] 松日楽 他: “ロボスタに作業を実行するための作業知能モジュール群の開発：プロジェクト概要と進捗”, 第26回日本ロボット学会学術講演会, 展開セッション9, 2008.
- [2] 河 他: “知能化環境構築のための位置管理モジュール及び環境サーバの設計”, 第26回日本ロボット学会学術講演会, RSJ2008AC1F2-08, 2008.
- [3] 野田 他: “減災情報共有データベースによる災害情報システム統合”, 第21回人工知能学会全国大会 予稿集, 人工知能学会, 宮崎, pp. 2E5-6 1-4, 2007.
- [4] 下羅 他: “災害時情報共有のためのプロトコルおよびその実装”, 人工知能学会 社会におけるAI研究会(第四回), pp. SIG-SAI-004-02 1-7, 2007.
- [5] 馬場 他: “ロボットタウンの実証的研究—タウンマネジメントシステムの開発—”, 第24回日本ロボット学会学術講演会予稿集 2N15, 9, 2006
- [6] OMG (Object Management Group): “The Robotic Technology Component Specification,” http://www.omg.org/technology/documents/domain_spec_catalog.htm