

在庫管理システムと自律移動配送ロボットを連携させた ラストワンマイルの自動化

実証実験レポート

2020.3.30 1.0版

TIS株式会社
会津大学
日本アドシス株式会社

目次：

1. 取り組みの背景と目的
 2. 実証実験概要
 - － 実証実験の目的
 - － 実施概要
 - － 実施体制と役割
 - － 使用したアーキテクチャー
 - － ロボット統合管理システムのソフトウェア構成
 - － 自律移動ロボットの使用機器、ソフトウェア構成
 3. 実証シナリオ
 4. 実証結果
 - － 検証項目
 - － 検証環境
 - － 検証結果（検証概要、検証結果、考察）
 - ① ERPシステムとの連携
 - ② 階層化された地図による自律移動手法の確立
 - ③ ロボットデータモデルとAPIの標準化
 5. まとめ
- 付録
- － メディア掲載
 - － 実証の様子
 - － アンケート結果

1. 取り組みの背景と目的

取り組みの背景と目的

IoT、AI、ロボットといった第4次産業革命の新たな技術革新は、人間の能力を飛躍的に拡張する技術であり、豊富なリアルデータを活用することで、様々な社会課題を解決できる。また、これらは大きな付加価値を生むものであるとして、新たな展開の「Society5.0」の実現が期待されている。

※「未来投資戦略2018－『Society5.0』『データ駆動型社会』への変革－」

背景

■サービスロボット活用への期待

サービス分野など非製造分野のロボット国内生産市場規模は、2020年では1.2兆円と試算※未来投資戦略2018

しかしながら、

特にサービス分野で、ロボットの社会実装は遅れている

■サービスロボット活用の課題

主に以下のような課題があげられる。

※経済産業省の考察等を参考に記載

- ・安価で汎用性のあるロボットが少ない
- ・インテグレーションのコストが高額
- ・導入を支援するインテグレータが不足している
- ・機械技術中心の技術開発で、クラウド等の多様な分野にわたる技術開発の広がりが無い
- ・サービスロボット活用を想定していない規制・制度に障壁がある

目的

ロボット技術、AI、IoTを利活用し「人とロボットの共生」を目指して、複数ロボットが協調できるプラットフォームを構築

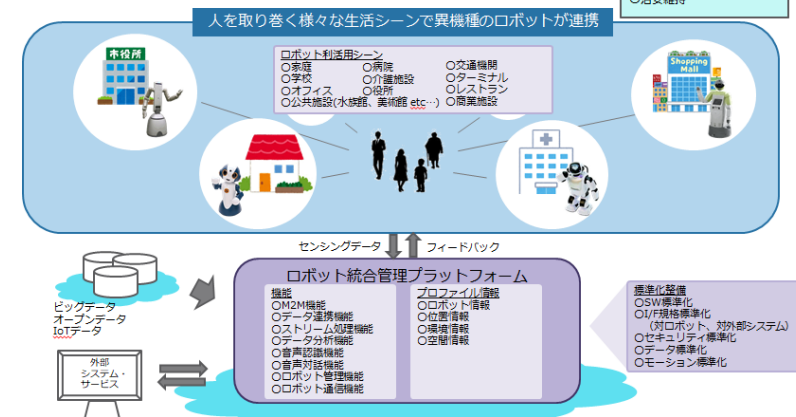
【実証実験目的】

1. 異種複数ロボット、IoT連携プラットフォームの機能および有効性検証
2. ロボットが取得したプライバシー情報（カメラ画像）の取扱い検証

人とロボットの共生

人とロボットがそれぞれの役割を果たしながら社会課題を克服し、QOLの高い社会を創成する。

社会課題
○少子高齢化
○人口減少
○過疎化
○持続可能な環境づくり
○教育レベルの向上
○治安維持



2. 実証実験概要

実証実験の目的

労働力不足や高齢化問題といった社会課題が顕著に配送業界において、業務の一部をサービスロボットに代行することが有益な手段の一つと考えている。
搬送業務のラストワンマイルの業務プロセスにサービスロボットを組み込んだユースケースを設定し、研究要素の確認を目的とした実証実験を実施する。

運送業界での課題

- ・公共交通機関が失われ、輸送能力が極端に低下
- ・労働力不足の深刻化（物流、清掃、警備、接客など）
- ・インターネット通販やECサイトの利用増加に伴い小口配送の増加

自律型モビリティ社会の実現
移動型サービスロボット活用に期待が高まる

人に寄り添いラストワンマイルの移動を支援する
サービスロボットによる課題解決を目指す

ロボットサービスへの期待

- ・エンタープライズシステムとサービスロボットの連携が容易でない
- ・異機種の搬送ロボットによるサービス提供に期待
- ・異機種のロボットを統合管理できるエコシステムに期待

多様なロボットサービスの普及
様々な業務でサービスロボットの活躍に期待が高まる

サービスロボットを業務システムに組み込むために
エコシステム（ロボット統合管理システム）の整備を目指す

実証実験の目的（研究要素）

1. ロボット統合管理システムとERPシステムとの連携の容易性の確認
2. 階層化された地図による自律移動手法の実現性の確認
3. ロボット統合管理システムと自律移動ロボット間のデータモデルの確認

実施概要

- 複数の業務システムと複数の自律移動配送ロボットを連携させた、ラストワンマイル自動化に係る実証実験

- 二種類の在庫管理システム

- クラウド在庫管理アプリケーション **ZAICO** ※1
- 在庫管理パッケージソフトウェア **EVEN** ※2

- 二種類の自律移動ロボット

- Turtlebot3** ※3 ベースの研究用ロボット
- メガローバー** ※4 ベースの研究用ロボット

- ロボット統合管理システム

- RoboticBase OSS版** ※5

- 場所と期間

会津大学「先端ICTラボ (LICTiA = リクティア)」 1F
2019/11/26 (火) -11/29 (金)



※1 : <https://www.zaico.co.jp/>

※2 : <https://www.tis-n.co.jp/it-solution/even/>

※3 : <http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>

※4 : <https://vstone.co.jp/products/wheelrobot/ver2.0.html#megarover2.0>

※5 : <https://github.com/RoboticBase>



実施体制と役割



TIS株式会社

ロボットを統合管理および制御するシステムのロボット統合管理システムの開発を担当

業務システムと連携し業務シナリオを実現

ロボットへの状態管理やタスク管理

ロボットの大局経路計画の実装



会津大学

自律移動ロボットのソフトウェアの設計と上位システム（ロボット統合管理システム）と自律移動ロボット間の役割の設計を担当

自律移動ロボットの局所経路計画の設計

自律移動ロボットの地図表現の設計

ロボット統合管理システムと自律移動ロボット間のデータモデルの設計



日本アドシス株式会社

自律移動ロボットのソフトウェア実装とハードウェア環境設定を担当

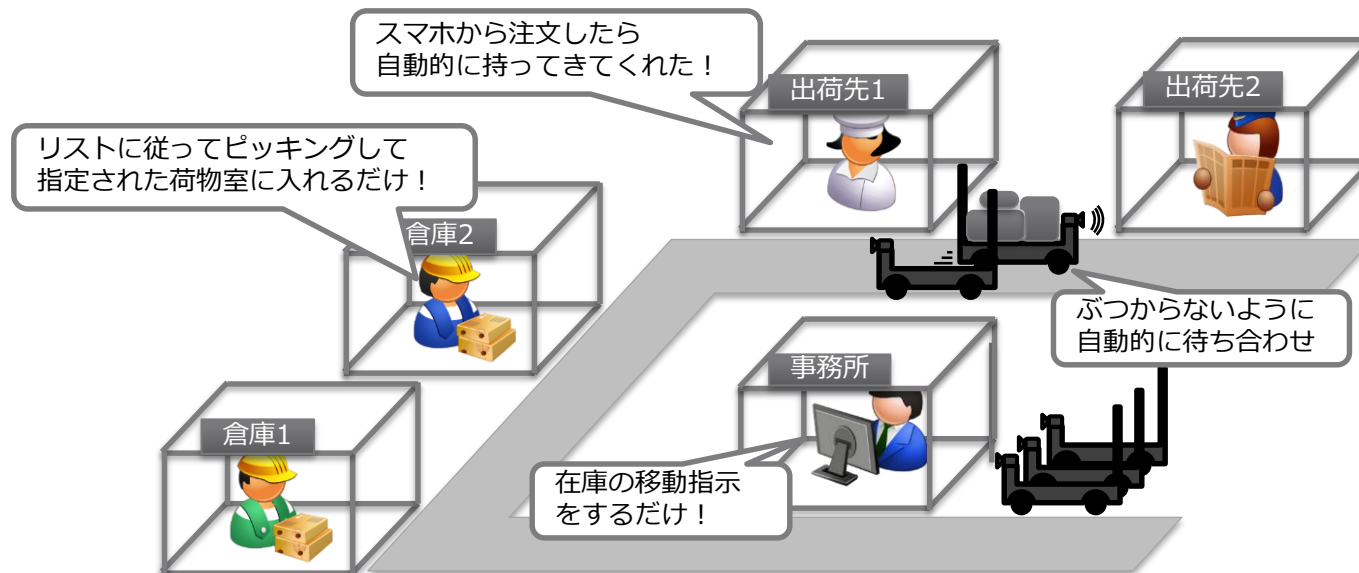
自律移動ロボット内で使用する地図システムの構築

自律移動ロボット内で局所経路計画の実装

自律移動ロボット（タートルボット、メガローバ）の走行環境設定

実証実験の内容

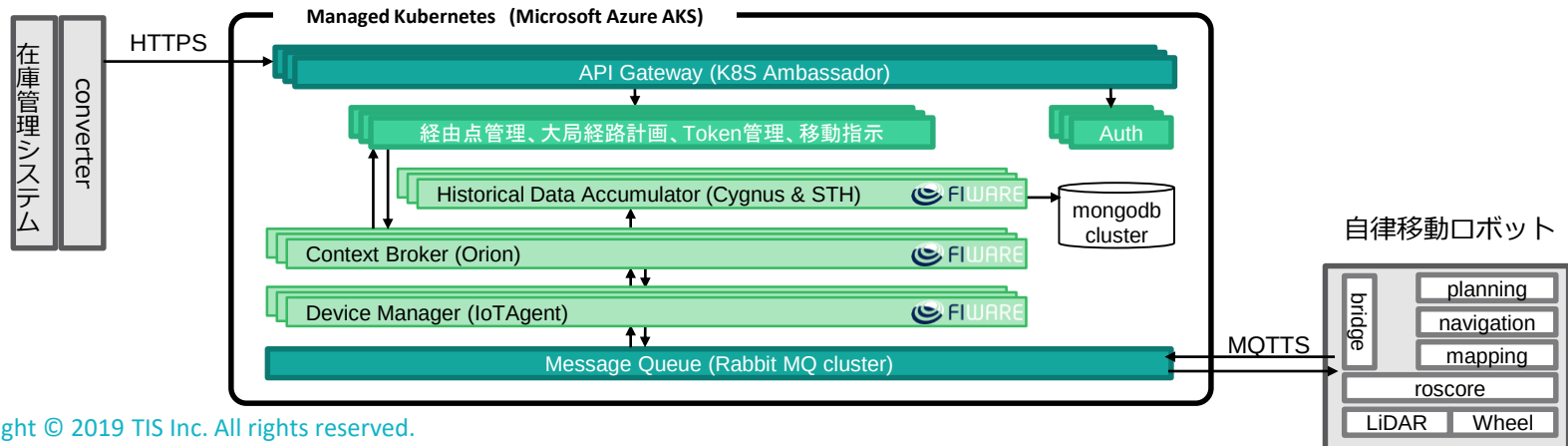
- ロボット統合管理システムによる集中管理
 - データモデルやAPIを標準化することにより、異なる業務システムや異なる自律移動ロボットを容易に管理可能
 - プラットフォームが仲介し振り分けることで、異なる業務を複数の自律移動ロボットが同時に遂行可能
 - プラットフォームが大局的な経路制御を行うことで、自律移動ロボットの衝突やデッドロックを防止



使用したアーキテクチャ

- ロボット統合管理システム
 - FIWAREを用いたオープンで標準化されたプラットフォーム
 - 目的地までの大局的な経路計画を有向グラフとして管理
 - 制限エリアへの侵入可否をTokenを用いて制御することで、複数ロボットの大局的な経路制御と移動指示を遂行
- 自律移動ロボット
 - ROSのLOAMで取得した三次元地図をロボットのセンサーの高さでスライスした二次元コストマップを保持
 - ROSのamclを活用した自己位置推定と、局所的な経路計画の遂行

ロボット統合管理システム



ロボット統合管理システムのソフトウェア構成

クラウド	クラウドサービス名称	バージョン等	備考
Microsoft Azure	AKS	1.14.6	
ソフトウェアカテゴリ	ソフトウェア名称	バージョン等	備考
FIWARE 	orion	2.2.0	
	Iotagent-json	1.10.0	
	cygnus-ngsi	1.15.0	
	sth-comet	2.5.0	
NoSQL DB	mongo DB	4.1.13	
Message Queue	RabbitMQ	3.7.16	
API Gateway	ambassador	0.73.0	
認証認可	fiware-ambassador-auth	0.3.0	
ロボット制御	uoa-poc2-controller	0.1.3	

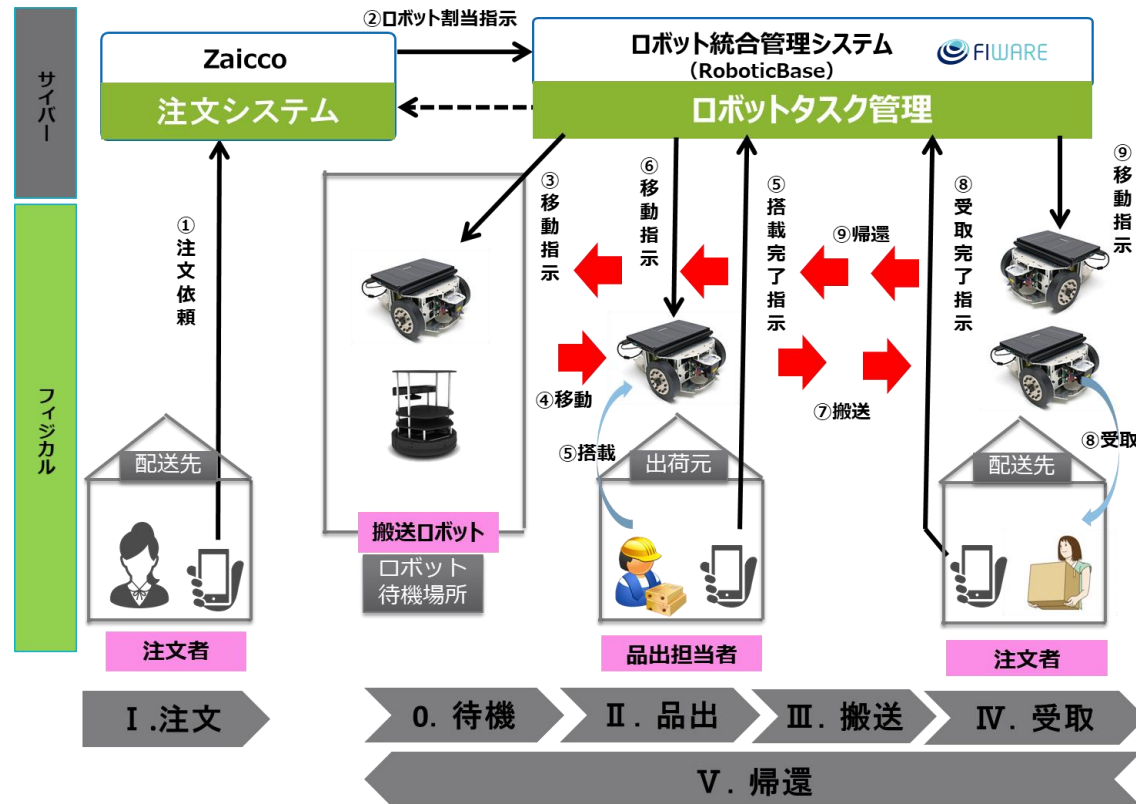
自律移動ロボットの使用機器、ソフトウェア構成

機器カテゴリ	機器名称	バージョン等	備考
自律移動ロボット	Turtlebot3	Version 1.0	PC名 : Intel NUC7i5BNH
	メガローバー	Version 2.0	
	Turtlebot3用BoxPC	CPU : Intel Core i5-7260U Memory : 16GB OS : Ubuntu 16.04 LTS	
センサー	メガローバー用BoxPC	CPU : Intel Core i3-7100U Memory : 16GB OS : Ubuntu 16.04 LTS	PC名 : ECS LIVA Z Plus
	Velodyne VLP-16	FPGA Version:3.0.40.0 F/W Version:3.0.40.0	
ユーザーインタフェース	Apple iPad mini 4	iPadOS 13.1.2	
ソフトウェアカテゴリ	ソフトウェア名称	バージョン等	備考
ロボット制御	ROS	Kinetic kame	
ユーザーインタフェース	Uoa-poc2-robotui	0.1.0	https://github.com/RoboticBase/uoa-poc2-robotui



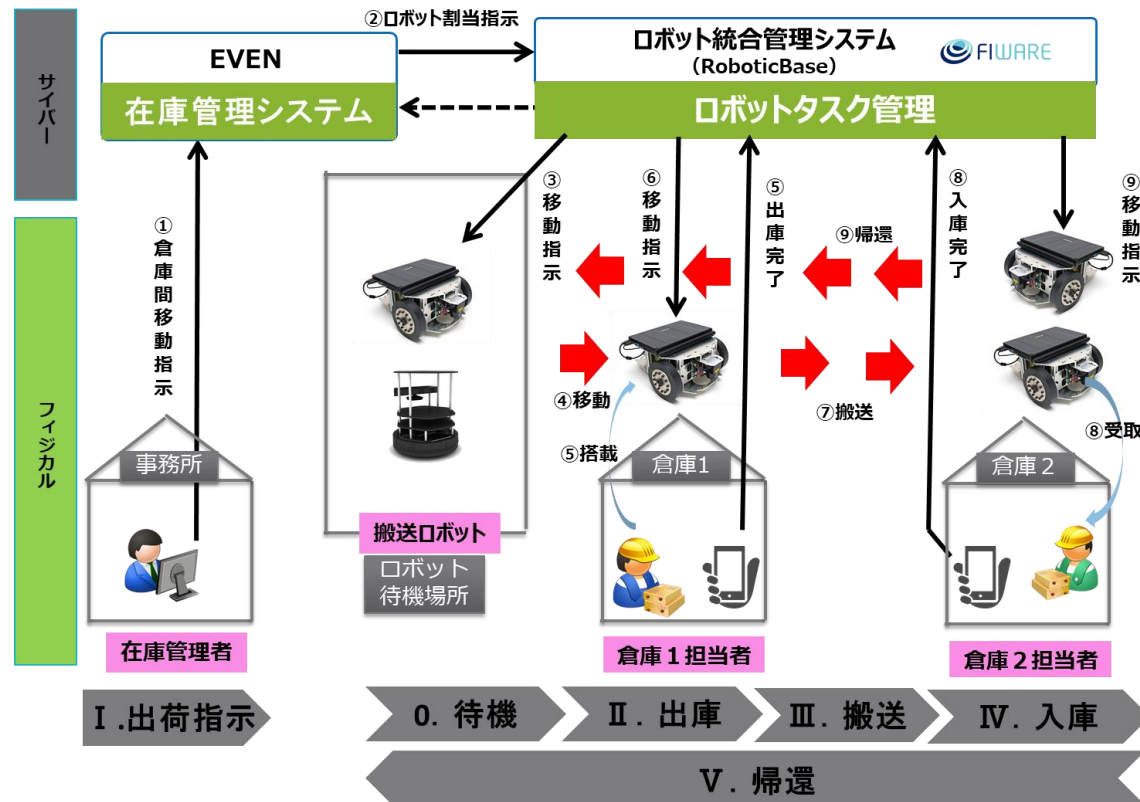
3. 実証シナリオ

シナリオ 1 – 注文から自動配送 –



- ① 顧客は、注文管理システムから品物を注文する
- ② 注文管理システムは在庫管理システム（zaico.co.jp）と連動し、ロボット統合管理システムへロボットの割り当てを依頼する
- ③ ロボット統合管理システムが適切なロボットを選出し、ロボットへ移動を指示する
- ④ ロボットが待機位置から指定された倉庫へ移動する
- ⑤ 倉庫の担当者がロボットへ荷物を積み、ロボットへ積込完了を指示する
- ⑥ 必要であれば、ロボットは異なる倉庫へ移動し、同様にして荷物を積み込まれる
- ⑦ 全ての荷物が積み込まれたら、ロボットは顧客の配送先まで移動する
- ⑧ 顧客が荷物の受け取りを確認すると、注文管理システムへ受け取りが通知される
- ⑨ 受取確認後、ロボットが待機位置まで移動する

シナリオ2 - 倉庫間移動 -



- ① 出荷担当が、在庫管理システム（EVEN）を用いて品物の倉庫間移動を指示する
- ② 在庫管理システムが、ロボット統合管理システムへロボットの割り当てを依頼する
- ③ ロボットプラットフォームが適切なロボットを選出し、ロボットへ移動を指示する
- ④ ロボットが待機位置から移動元倉庫へ移動する
- ⑤ 倉庫の担当者がロボットへ荷物を積み、ロボットへ積込完了を指示する
- ⑥ 必要であれば、ロボットは異なる倉庫へ移動し、同様に荷物を積み込まれる
- ⑦ 全ての荷物が積み込まれたら、ロボットは移動先倉庫まで移動する
- ⑧ 倉庫の担当者が荷物の受け取りを確認すると、ロボット統合管理システムへ受け取りが通知される
- ⑨ 受取確認後、ロボットが待機位置まで移動する

4. 実証結果

検証項目

① ERPシステムとの連携

ロボット統合管理システムが企業のERPシステムとAPIを通じて連携し、ERPシステムと自律移動ロボットのギャップを埋めることで、様々な種類の自律移動ロボットを企業の経済活動へ容易に組み込めるようにすること。

② 階層化された地図による自律移動手法の確立

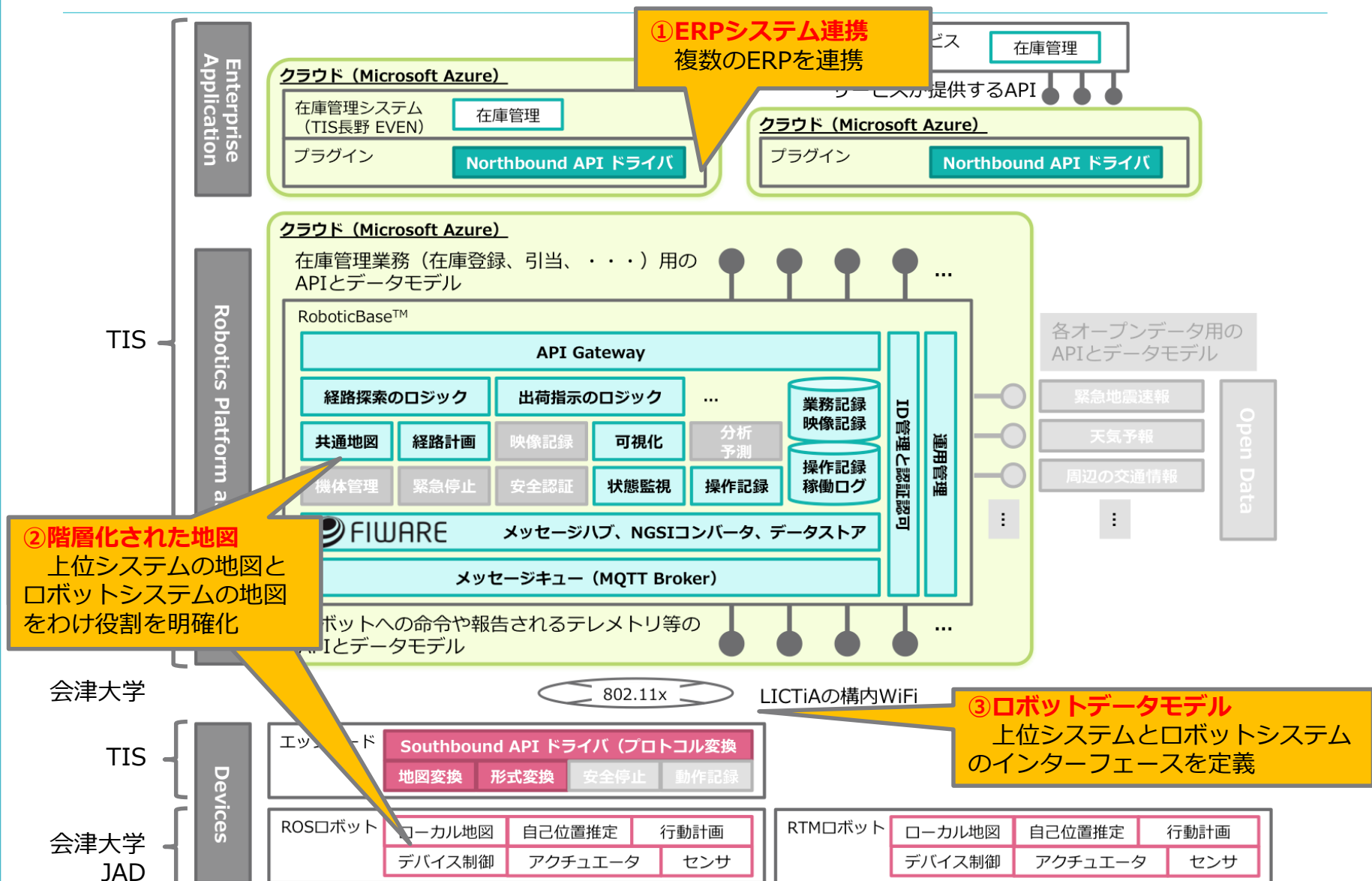
自律移動ロボットの内部情報としての地図（エゴ地図）と、自律移動ロボットの機械的特性に縛られない客観的共通知識としての地図（ソシオ地図）、及びヒトが見て理解できる地図（見取り図）といった多層構造の地図という概念と、それらを相互変換し「自律移動ロボット用のダイナミックマップ」として機能させる技術を確認すること。

更に上位系システム（ロボット統合管理システム）にてロボット単体で認識が難しい情報を認識したうえで、複数ロボットの運行管理する技術を確認すること。

③ ロボットデータモデルとAPIの標準化

業務に特化したソフトウェアが、様々な種類の自律移動ロボットの機械的特性を意識せずに連携できるように、自律移動ロボットの機種に依存しない抽象的なデータモデルとAPIを整備し標準化が可能にすること。

検証環境 実証実験全体アーキテクチャと検証項目

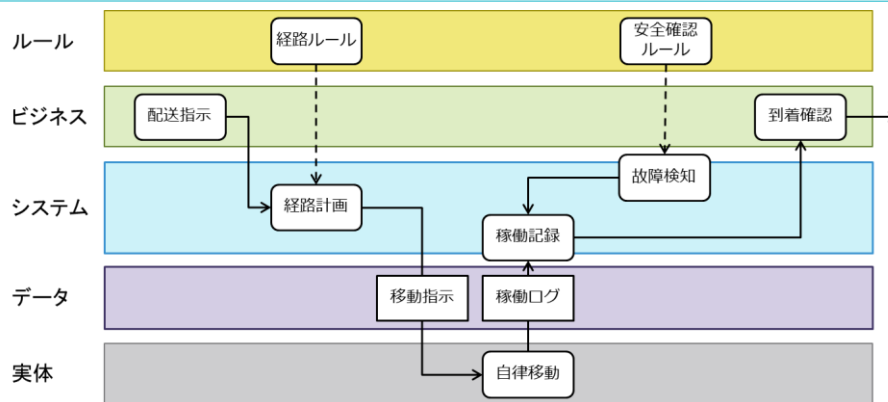


検証結果：

① ERPシステムとの連携 【概要】

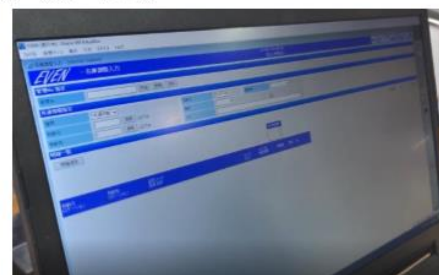
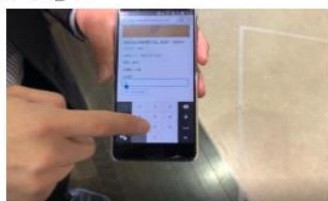


商品配送申込



サービスロボットを含む業務プロセス(例)

出荷指示 在庫管理システム



商品の在庫が連動して増減
商品の追跡なども可能

集荷作業 受領



商品受取用QR

ロボットが倉庫（店舗）を経由し配送先へ移動

検証結果：

① ERPシステムとの連携 【使用 E R P システム】

ERPシステム（在庫管理）については、パッケージソフトの**EVEN**およびクラウド在庫管理ソフトの**zaico**を使用

◆ EVEN

- 倉庫間移動のシナリオで利用
- ノート P C を利用して V M 上に EVEN サーバを構築し、ロボット統合管理システムへの接続用 Proxy サーバを用意
- 在庫調整機能(在庫の移動)のみを連携

◆ zaico

- 注文配送のシナリオで利用
- WebAPI を利用してロボット統合管理システムと接続
- フロントエンドはスマートフォンのブラウザに対応

検証結果：

① ERPシステムとの連携 【使用 E R P システム比較】

EVENとzaicoクラウドサービスの在庫管理比較まとめ

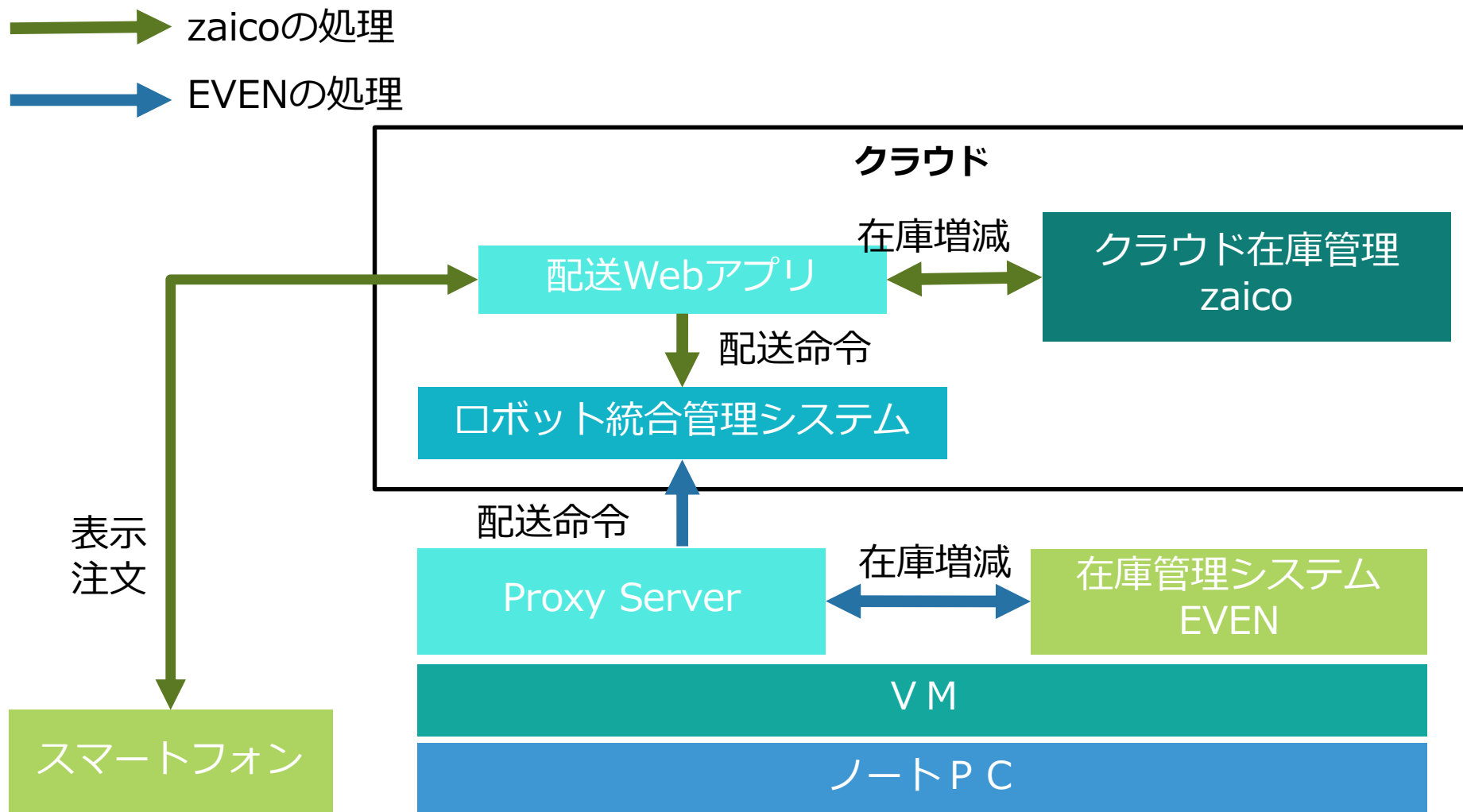
システム	システム提供形態	U I	API提供	運用管理	業務適用
EVEN URL: https://www.tis-n.co.jp/it-solution/even/	パッケージソフトとして導入	ある	なし	自身でシステムの管理が必要	業務フローに特化
zaico URL : https://www.zaico.co.jp/	クラウドサービスとして提供（月額提供）	ある	あり	サービスなのでシステムの管理は不要	自由カスタマイズ必要

参考)
TIS長野出展 T I S と会津大学が自律移動するロボットを活用した搬送業務自動化の実証実験に、T I S 長野のソリューション「在庫管理システムEVEN」を活用

URL : <<https://www.tis-n.co.jp/news/news-20191223/>>

検証結果：

① ERPシステムとの連携 【システム構成図】



検証結果：

① ERPシステムとの連携 【課題と考察（EVEN）】

項番	実施時の問題点や課題	考察	備考
1	完全な在庫管理業務フローの再現は外部インターフェイスなどを必要とするため実現できなかった	EVENに限らずERPとロボット統合管理システムを接続するためにはERP側に接続用の実装が必要であると確認	
2	在庫移動指示からロボットへの配送命令は安定して動作していた（1度だけネットワークが不安定だったため配送命令失敗）	ERP側にコールバック処理を登録できるような仕組みがあれば連携が容易	
3	EVENが正常に動作するブラウザ環境がIE5～9のためjavascriptから直接ロボット統合管理システムへHTTPS接続することが困難だった	EVEN仕様のため受容	
4	一度だけノートPCがブルースクリーンによって強制終了した（MEMORY MANAGEMNT）	本質でないため再現待ち	

検証結果：

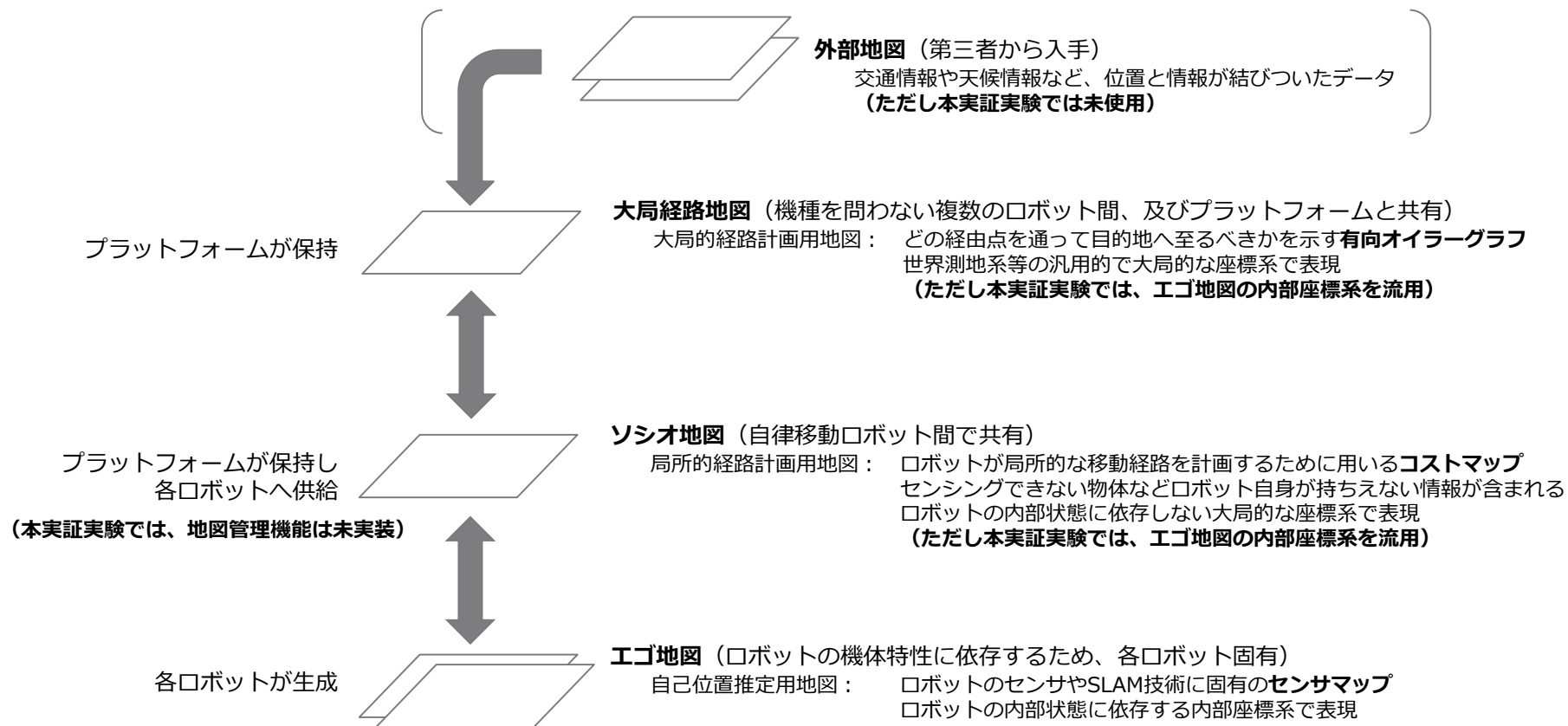
① ERPシステムとの連携 【課題と考察（z a i c o）】

項番	実施時の問題点や課題	考察	備考
1	在庫管理を行うだけのAPIのため注文画面の実装が別途必要だった	一般的なREST APIだったため EVENに比べて接続はしやすい	
2	本質ではないUIの実装に苦戦することが多々あった	シナリオに対応したUIを実装するためにコストがかかる	
3	一般的なREST APIのためトランザクション処理に対応していなかった	W e b アプリでトランザクション処理を対応にて対処する	

検証結果：

② 階層化された地図による自律移動手法の確立 【概念】

【階層化した「地図」の概念と役割】



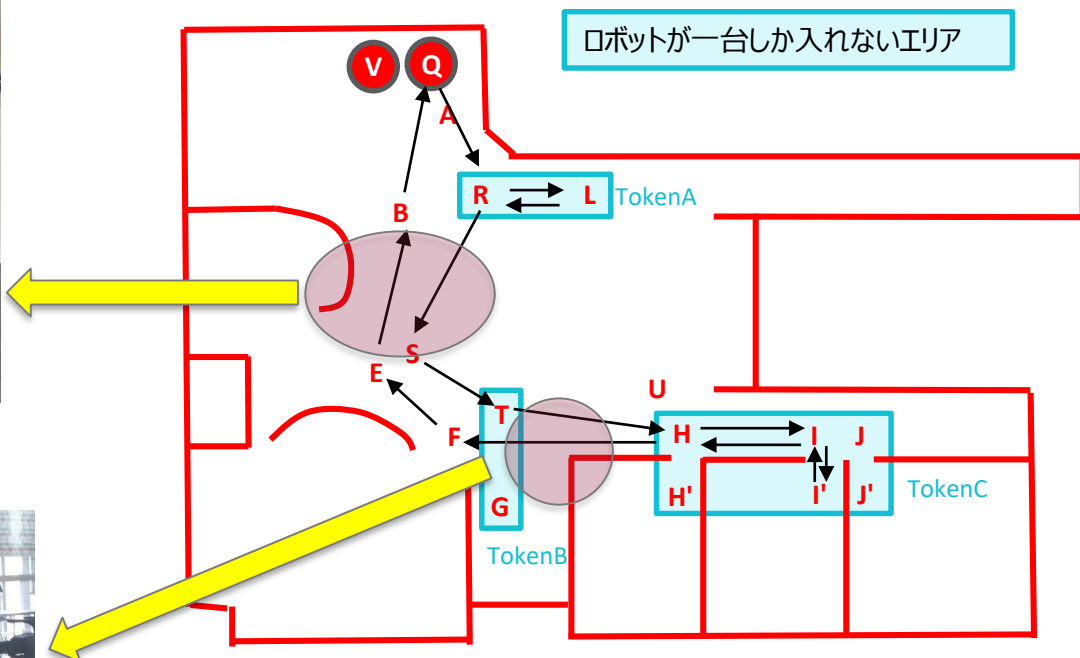
検証結果：

② 階層化された地図による自律移動手法の確立 【大局経路地図概要】

【大局経路地図の概要】



広いスペースではロボットはすれ違い可能



自律移動ロボットが倉庫をめぐって目的地へ至り、出発点へ戻ってくる
大局経路な経路計画をロボット統合管理システムが有向オイラーグラフ
として保持

閉塞区間として定義された狭いスペースでは、先行ロボットが
閉塞区間内から脱出するまで後続ロボットは待機

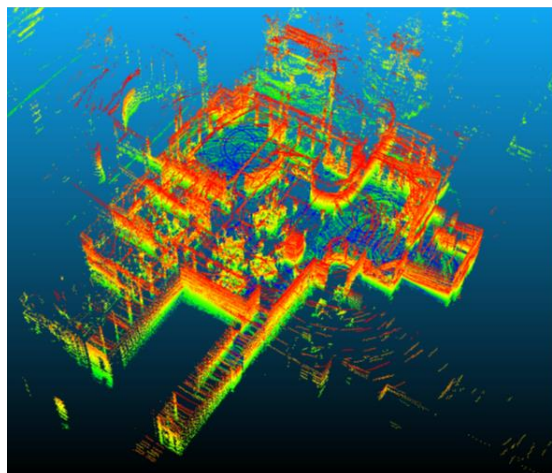
検証結果：

② 階層化された地図による自律移動手法の確立 【エゴ地図ソシオ地図】

【エゴ地図とソシオ地図の関係（1 / 2）】

- 今回三次元地図を使用したナビゲーションアルゴリズムは開発が間に合わなかったため、二次元ナビゲーションを採用した。三次元点群地図を二次元占有格子地図変換したものをエゴ地図とし、センサーでは見えない物体（ガラス等）を追記したものをソシオ地図として使用

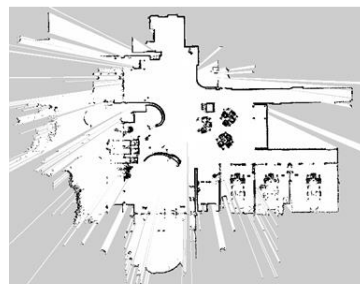
【3D点群地図】



2D変換



【エゴ地図】



ガラス追加・雑点除去



【ソシオ地図】



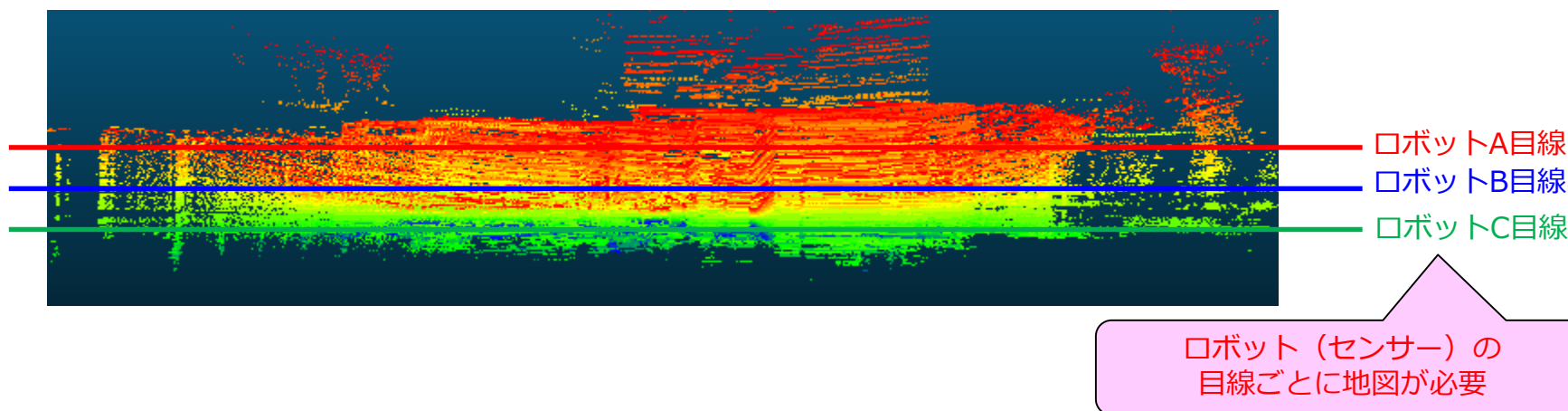
ガラス部分（赤線）を
手動追加

検証結果：

② 階層化された地図による自律移動手法の確立 【エゴ地図ソシオ地図】

【エゴ地図とソシオ地図の関係（2 / 2）】

- 二次元占有格子地図をエゴ地図/ソシオ地図としたため、ロボットのセンサーの高さ/傾きが異なる場合、ロボット毎に地図を作成する制約が出てしまった（今回の実証実験では複数ロボットのセンサーの高さ/傾きを合わせ、共通の地図で対応）
- 今後三次元ナビゲーションを使用した、上記制約を廃した環境での継続的な検証が必要



検証結果：

② 階層化された地図による自律移動手法の確立 【課題と考察】

項番	実施時の問題点や課題	考察	備考
1	ロボットのHWに関する考慮漏れが起因となった自己位置喪失が発生した	HW仕様の把握とノウハウ共有を図る	
2	三次元点群をエゴ地図へ変換するのは時間がかかり、またエゴ地図へセンシングできない物体を反映しソシオ地図化する作業は手作業で行っていたため、ノウハウが必要で手間がかかった	エゴ地図・ソシオ地図作成用のUIを準備することで、地図の維持管理を簡便化を検討	
3	大局的経路のデータ登録も手作業で行っていたため手間と時間がかかった	大局経路のデータ登録用のUIを準備することで、地図の維持管理を簡便化を検討	
4	エゴ地図、ソシオ地図を更新する際には、ロボットにリモートログインして地図を入れ替える必要があった	ロボット統合管理システムによるロボットへのエゴ地図・ソシオ地図のリモート配信と、ロボットが持つ地図の動的入替を可能な仕組みの検討	

検証結果：

② 階層化された地図による自律移動手法の確立 【課題と考察】

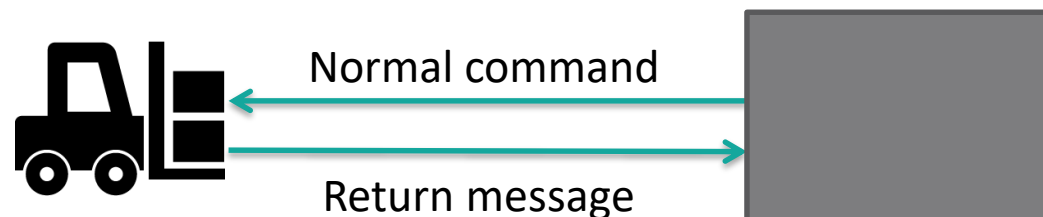
項番	実施時の問題点や課題	考察	備考
5	大局経路の有向オイラーグラフは静的なため、経路のノード上に障害物があるとロボットがノードに到達できずに立ち往生することがあった	外部地図やロボットの観測データ、監視カメラの情報等を活用し、大局経路を動的に生成する仕組みの検討	
6	あるロボットで作成したエゴ地図の内部座標系を大局経路地図の座標系に流用したため、エゴ地図の再作成が大局的経路の設定にも影響を及ぼした	エゴ地図・ソシオ地図・大局経路地図・外部地図それぞれで適切な座標系を選択できるようにし、それぞれの地図間で座標系を自動変換できる仕組みの検討	

検証結果：

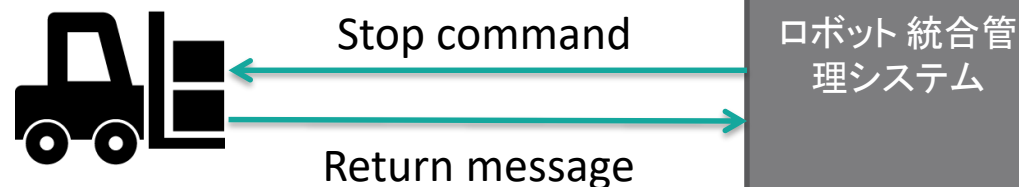
③ ロボットデータモデルとAPIの標準化 【作成したデータモデル】

【自律移動ロボット用データモデル】

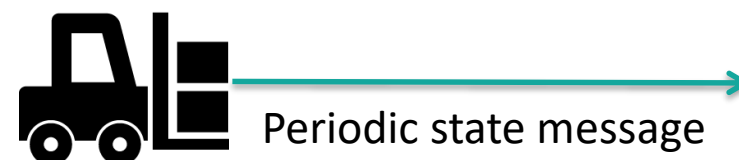
i. ロボットへの移動指示と受信結果



ii. ロボットへの停止命令と受信結果



iii. ロボットの状態報告



検証結果：

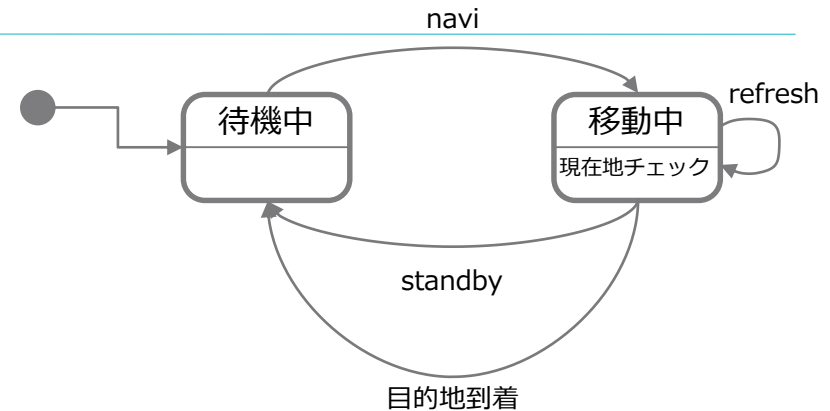
③ ロボットデータモデルとAPIの標準化

【データモデル詳細】

i. ロボットへの移動指示と受信結果

i - a 移動指示

Name	Type	Description	
id	string	Unique ID of the robot	
type	string	Entity name	
time	string	Send date (YYYY-MM-DDThh:mm:ss.sssZ)	
command	string	Command: ["navi", "standby“, “refresh”]	
waypoints	array	List of waypoints	
	point	object	Coordinates of each waypoint
	x	number	x-coordinate
	y	number	y-coordinate
	z	number	z-coordinate
	orientation	object	Orientations of each waypoint
	roll	number	Roll
	pitch	number	Pitch
	yaw	number	Yaw



- ROSの場合、個々のロボットごとに専用のROS Topicを設定する
- “time”のフォーマットは、ISO8601拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ)
- “cmd”は [“navi”, “refresh”, “standby”] のいずれか
- “waypoints” (waypointのリスト) に上限数は無い
- 最後のwaypointが目的地点を意味し、目的地点まで到達すればロボットは自律的に停止し待機状態になる
- 待機中に“navi”を受け取ると、指定された“waypoints”に従って自律的に移動する
- 自律移動中に“reflesh”を受け取ると、以降のwaypointsを破棄し、受領した新たなwaypointsに基づいて自律移動を継続する
- 自律移動中に“standby”を受け取ると、その場で停止して待機状態になる
- 与えられたwaypointsにangleが含まれない場合は、通過姿勢は指定されず任意の姿勢で通過してよい。一方angleが含まれる場合は、指定されたpoint通過時にangle.valueの方向を向く。

検証結果：

③ ロボットデータモデルとAPIの標準化 【データモデル詳細】

i. ロボットへの移動指示と受信結果

i - b 受信結果

Name	Type	Description
id	string	Unique ID of the robot
type	string	Entity name
time	string	Send date
receivedTime	string	Received date
receivedCommand	string	Received command
receivedWaypoints	array	Received waypoints
point	object	Coordinates of each waypoint
x	number	x-coordinate
y	number	y-coordinate
z	number	z-coordinate
orientation	object	Orientations of each waypoint
roll	number	Roll
pitch	number	Pitch
yaw	number	Yaw
result	string	Reaction of the command
errors	array	Error messages

- “time”のフォーマットは、ISO8601拡張形式+ミリ秒（YYYY-MM-DDThh:mm:ss.sssZ）
- “received_cmd”等は、受信した “cmd” 等を送り返す
- “result”は [“ack”, “ignore”, “error”] のいずれか
- 何らかのエラーが発生した場合は、その理由を文字列として “errors” のリストに追加する（複数可）

		navi	refresh	standby
待機中	正常	ack	ignore	ignore
	異常発生	error	-	-
移動中	正常	ignore	ack	ack
	異常発生	-	error	error

検証結果：

③ ロボットデータモデルとAPIの標準化 【データモデル詳細】

ii. ロボットへの停止命令と受信結果

ii – a 停止命令

Name	Type	Description
id	string	Unique ID of the robot
type	string	Entity name
time	string	Send date
emergencyCommand	string	Emergency command

- ROSの場合、個々のロボットごとに専用のROS Topicを設定する
- “time”のフォーマットは、ISO8601拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ)
- この緊急停止メッセージは移動指示メッセージとは独立したプロセスとして処理され、どのような状態であれ、このメッセージを受け取ったらロボットはその場で安全に停止する

ii – b 受信結果

Name	Type	Description
id	string	Unique ID of the robot
type	string	Entity name
time	string	Send date
receivedTime	string	Received date
receivedEmergencyCommand	string	Received emergency command
result	string	Result for the command
errors	array	Error messages

- “time”のフォーマットは、ISO8601拡張形式+ミリ秒 (YYYY-MM-DDThh:mm:ss.sssZ)
- “received_cmd”等は、受信した “cmd” 等を送り返す
- “result”は **[“ack”, “error”]** のいずれか
- 何らかのエラーが発生した場合は、その理由を文字列として “errors” のリストに追加する（複数可）

検証結果：

③ ロボットデータモデルとAPIの標準化 【データモデル詳細】

iii. ロボットの状態報告

Name	Type	Description
id	string	Unique ID of the robot
type	string	Entity name
time	string	Send date
mode	string	Command
errors	array	Error messages
pose	array	Current position
point	object	Coordinates of the position
x	number	x-coordinate
y	number	y-coordinate
z	number	z-coordinate
orientation	object	Orientations of the position
roll	number	Roll
pitch	number	Pitch
yaw	number	Yaw
destination	array	Current destination
point	object	Coordinates of the destination
x	number	x-coordinate
y	number	y-coordinate
z	number	z-coordinate
orientation	object	Orientations of the destination
roll	number	Roll
pitch	number	Pitch
yaw	number	Yaw

covariance	string	Error covariance matrix of estimated position
battery	array	Battery states
voltage	object	Voltage
current	object	Current

- ROSの場合、個々のロボットごとに専用のROS Topicを設定する
- ロボットが起動している間は、定期的に送り続ける（毎秒1回程度）
- “time”のフォーマットは、ISO8601拡張形式+ミリ秒（YYYY-MM-DDThh:mm:ss.sssZ）
- “mode”は “navi”, “standby”, “error” のどれか
- ロボットが待機中の場合は “standby” を、自律移動中の場合は “navi” を、何等かのエラーが発生した場合は “error” を送る
- 何らかのエラーが発生した場合は、その理由を文字列として “errors” のリストに追加する（複数可）
- 位置推定の精度は、共分散行列（36要素の数値列）として報告する
- 同一方向に1回転以上した場合、angleの要素に2π以上（あるいは-2π以下）の値を入れても良い
- currentの値が取得できない場合は、currentの値を省略してよい

検証結果：

③ ロボットデータモデルとAPIの標準化 【課題と考察】

項番	実施時の問題点や課題	考察	備考
1	本実証実験シナリオに沿ってデータモデル化しているため、自律移動ロボットを使用した他のユースケースでの利用は想定されていない	遠隔操作や人追従など配送以外のユースケースを想定したモデル化を検討	
2	地図情報などはロボット間でのデータ共有が考えられるが、データモデル化されていない	地図情報などサービスロボット全般で共有されるデータのモデル化を検討	
3	サービスロボット全般のデータモデル化が目標であるが、今回は自律移動ロボットのみデータモデル化した	アームロボットなどの他サービスロボットを使用したシナリオでのデータモデル化を検討	
4	本データモデルを国際標準として広く認知させるには、FIWAREだけでなくISOなどの国際標準への登録が必要	FIWARE Foundationや国内各団体との連携によるデータモデルの標準化活動に参加	
5	データモデルの精査には、他での利用とその知見の収集が必要	ロボットメーカーとの意見交換によるデータモデルの精査を実施	

5. まとめ

まとめ：① ERPシステム連携

実施できたこと

- ◆ ERPシステムと連携する業務シナリオを整理し、「配送ロボットの移動」というタスクを取り出してロボット統合管理システムが提供するAPIとデータモデルに合致する抽象化レイヤを挟むことで、異なるERPシステムの異なる業務シナリオを同一の自律移動配送ロボットに担務させることができた。
- ◆ 異なるERPシステムでも問題なく連携することができた。

課題や問題点

- ◆ APIとデータモデルの抽象度を上げすぎると、個別の業務シナリオへの対応が困難になる。
例) 注文配送時と倉庫間移動時でのロボットの発話内容の変更
→ 本実証実験では、対人インターフェースはロボットとは切り離してiPadで実装し、自律移動のAPIとは別にして、個別の業務シナリオへ対応
- ◆ 各ERPシステムとの連携には、個別対応が必要なケースが多い。

今後の活動方針

- ◆ 実務に即した経験の蓄積によるAPIやデータモデルの更改
- ◆ ロボットのコア機能と付加機能の切り分け

まとめ：② 階層化された地図による自律移動と経路計画

実施できたこと

- ◆ ロボットのセンサー（LiDAR）が認識できないガラスの壁等を手動でコストマップに反映することで、ロボット自身が周囲を観測し得ない場所であっても自律的に移動することができた。
- ◆ ロボットが倉庫や目的地を巡る大局的経路を有向オイラーグラフとして表現し、単線区間（ロボットが一台しか入れないエリア）を設定してTokenによる閉塞管理を行うことで、ロボット同士の衝突やデッドロックを防止することができた。

課題や問題点

- ◆ ロボットのHWに関する考慮漏れが起因となった自己位置喪失が発生した。
- ◆ コストマップは静的で、かつセンシングできない物体の反映は手作業で行う必要があったため、実験環境の変化に追従させるには手間と時間がかかった。
- ◆ 大局的経路の有向オイラーグラフは静的で、手作業で設定する必要があったため、新たな経路を登録するには手間と時間がかかった。
- ◆ 各層の地図は座標系を共有しており、コストマップを再作成し座標系が変更されると大局的経路の設定にも影響が及ぶため、コストマップの再作成ができなかった。

今後の活動方針

- ◆ コストマップ修正の簡便化
- ◆ 外部地図やロボットの観測データを活用した、最適な大局経路の動的生成
- ◆ 大局経路地図、ソシオ地図、エゴ地図間の座標系変換

まとめ：③ 標準化を目指したデータモデル

実施できたこと

- ◆ 自律移動に関するデータモデルを整理し、標準的なデータモデルの案としてFIWARE Foundation や業界団体に提案できた。
- ◆ ロボット革命イニシアティブ（RRI）の調査検討委員会に、当データモデルを持ち込み方向性が正しいことを確認できた。

課題や問題点

- ◆ 本実証実験以外での有効性は不確かである。
- ◆ バッテリー状態の通知など、自律移動ロボットの運用管理に必要なデータの選別とその送受信は検証できなかった。
- ◆ 今回は、搬送ロボットを使ったユースケースに絞ったデータモデルのみの実装であったが、他のユースケースについての確認ができなかった。

今後の活動方針

- ◆ 実務に即した経験の蓄積によるデータモデルの更改
- ◆ 各諸団体と連携して標準化活動への貢献
- ◆ 他のユースケースについても、我々の取りくむ範囲を見定めながら検討

付録

メディア掲載

搬送ロボットを活用した実証実験の様子。荷物を積んだ手前と右奥の搬送ロボットが施設内を自律走行する



ロボット、荷物運べた？

自律走行する搬送ロボット 目指す。
による荷物運搬の実証実験が、実験は会津大とIT企業「TIS」などが共同で実施。26日、会津若松市の会津大で始まった。28日まで行われ、搬送ロボットはレーザー距離計を搭載しており、施設内の壁や別の搬送ロボットとの距離の解消に向けた技術の確立を

会津大が実証実験 * あすまで一般公開

離間を検知することで、ロボット内部で作製した地図を頼りに人や障害物をよけながら自律走行ができる。荷物の注文にはスマートフォン、受け取りにはQRコードを活用している。

TISの担当者によると、実用化が進めば、一つの施設内で荷物運搬や警備、清掃などさまざまな機能を持つロボット同士が連携し、新たなサービスを展開することが可能になるという。歴代東会津大特任教授は「今後は屋外での実証も可能にするため、ハードとソフトの両面でシステム改良を進めていきたい」と話した。

実験は会津大内の先端ICTラボ「LICTIA」(リクティア)で一般公開している。時間は午後1時~同5時。問い合わせは会津大復興支援センター(電話0242・37・2500)へ。

2019年11月27日 福島民友 朝刊

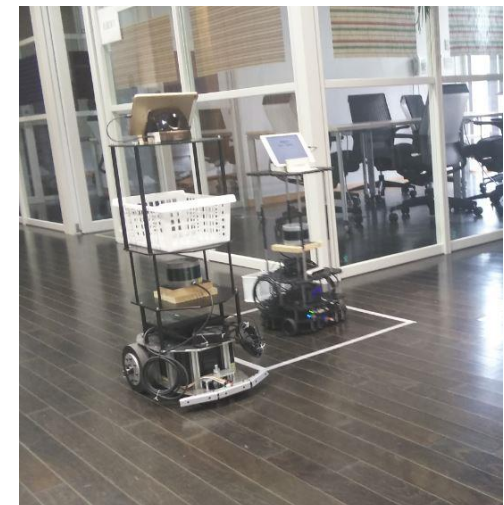
他、下記新聞紙面にも掲載いただきました。

2019年11月27日 福島民報 朝刊

2019年11月27日 日刊工業新聞 朝刊

2019年11月27日 読売新聞(福島地方版) 朝刊

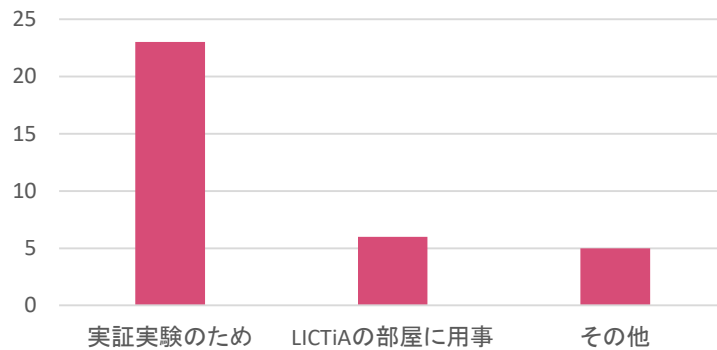
実証の様子（写真）



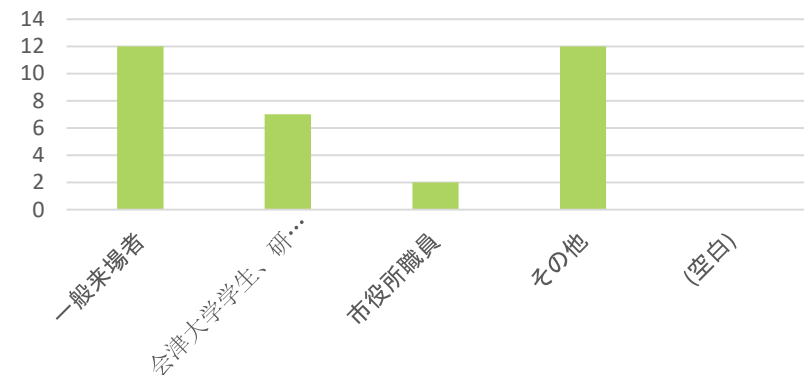
アンケート結果（訪問者の属性）

- 回答人数： 34名

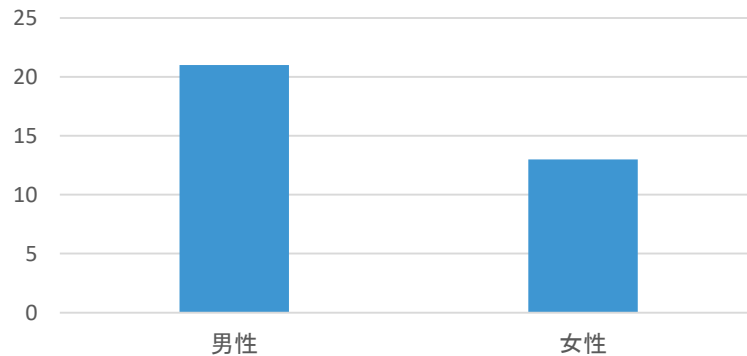
LiCTiA来訪目的



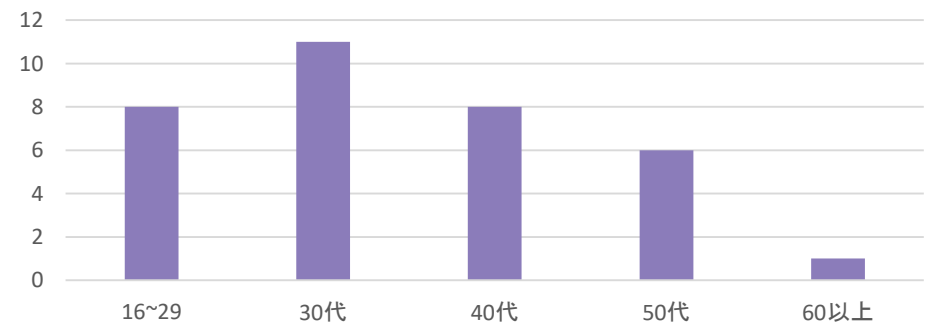
訪問者の属性



訪問者の性別

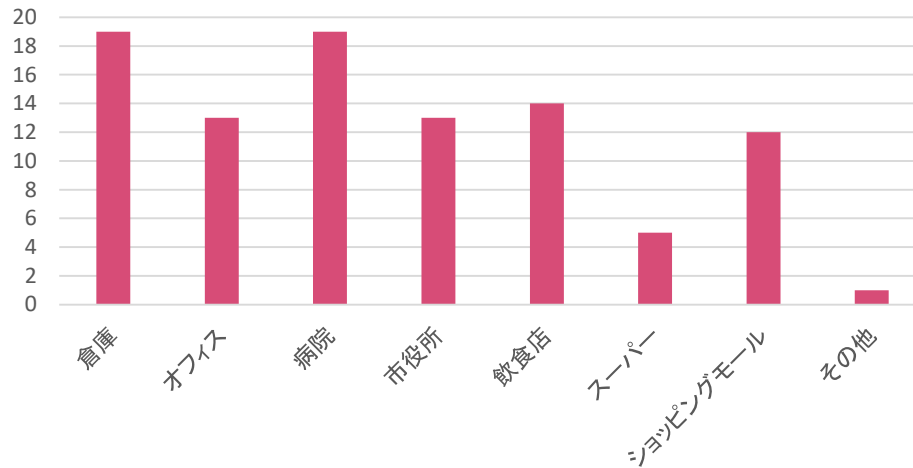


訪問者の年代

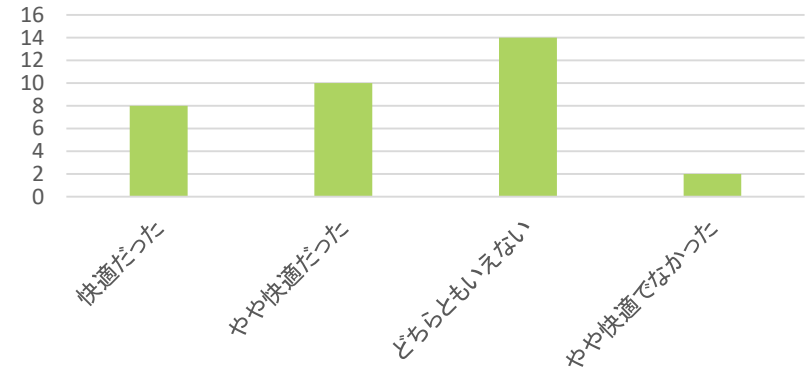


アンケート結果（訪問者の印象）

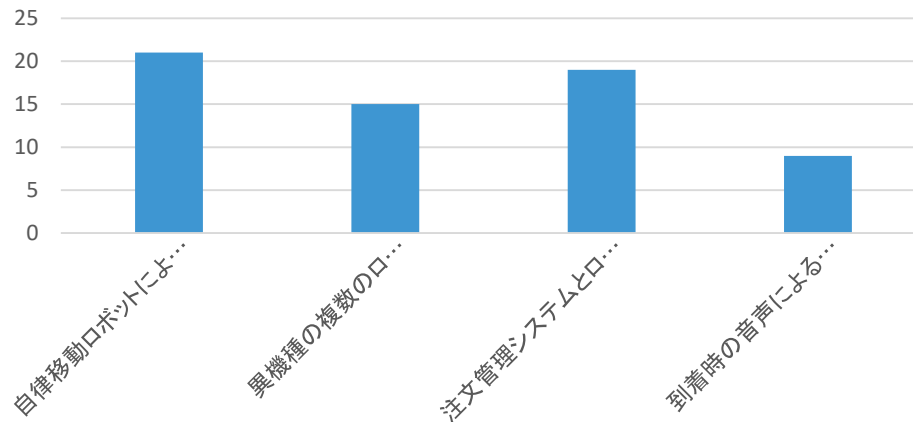
あると便利な場所



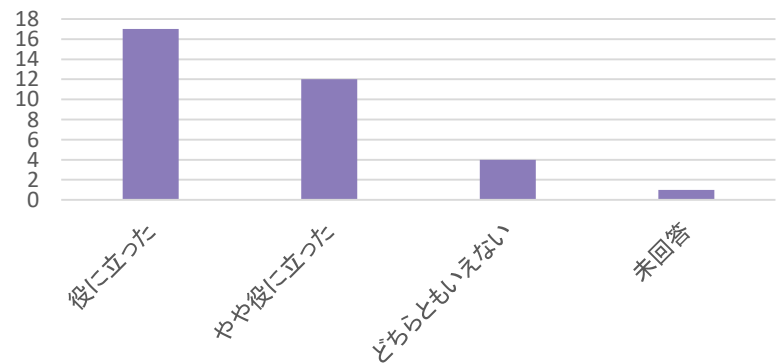
快適に感じたか



魅力を感じた部分

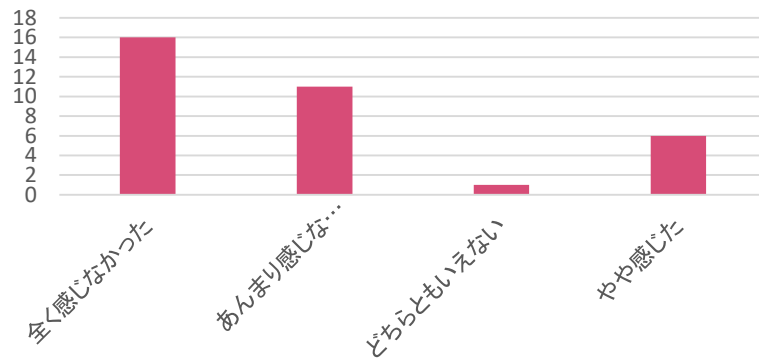


役に立つか

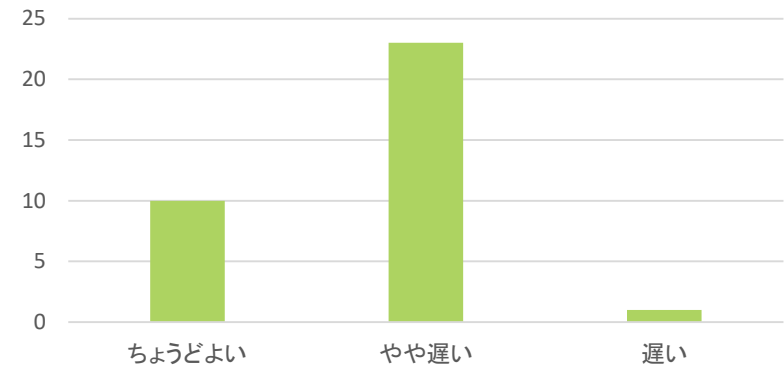


アンケート結果（ロボットの速度関連）

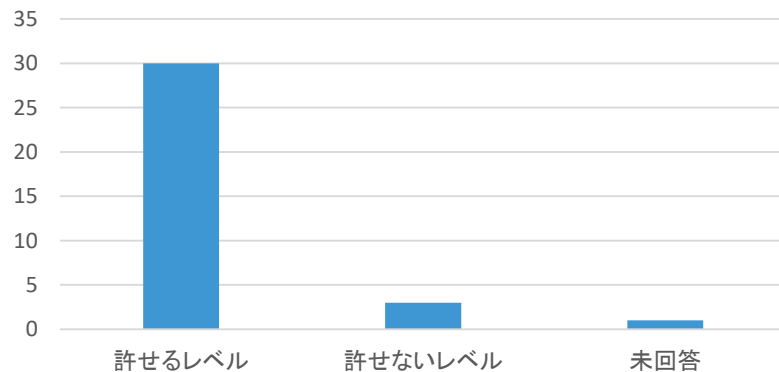
ロボットの移動に危険を感じたか



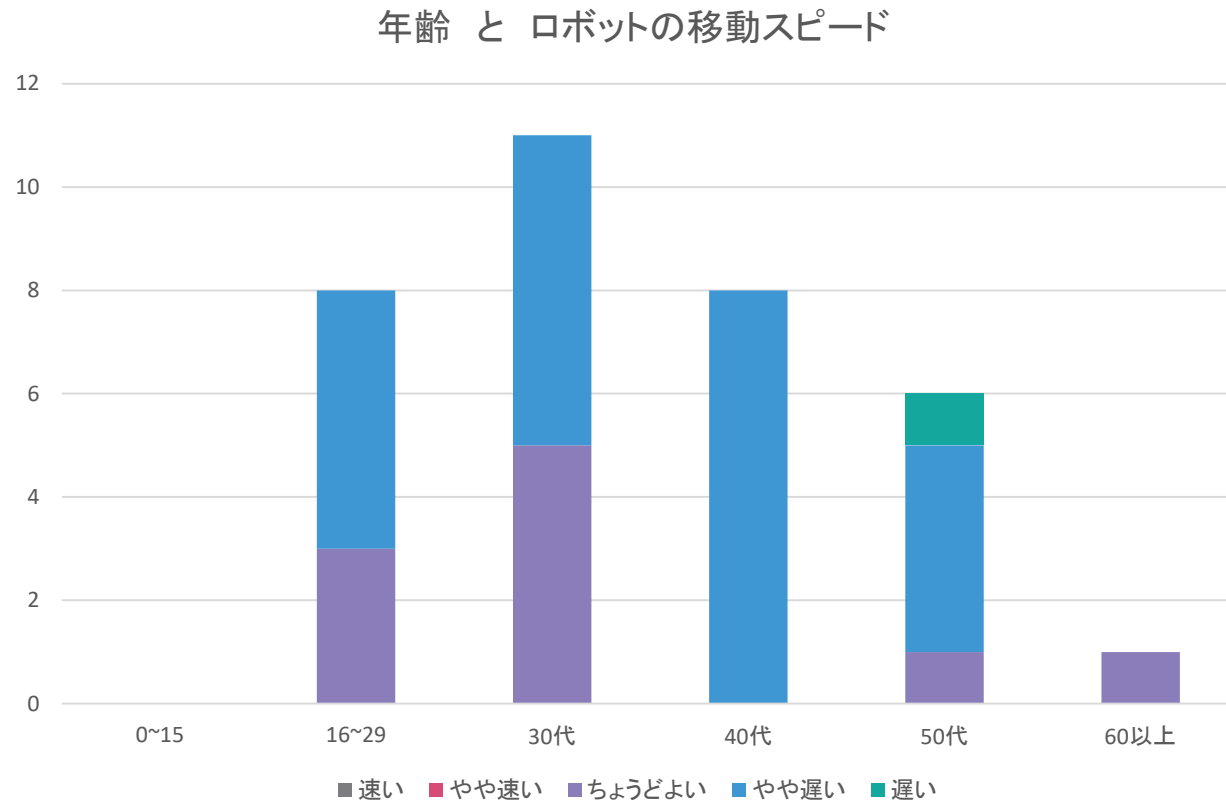
ロボットの移動スピード



移動スピードは許せるか



アンケート結果 (相関)



アンケート結果 主なコメント

- 快適に感じた/感じなかった 理由
 - 人に会いたくない人は、ロボットのほうが良いかもしれません。
(やや快適だった)
 - 人のほうが早いのでまだロボットである理由が乏しい (やや快適でなかった)
 - 時間の短縮、受け取りの確認が確実 (やや快適だった)
 - 動作を見ていると位置をみるために時々回転している。その回転が多いと大丈夫かと不安になる (どちらともいえない)
 - 同じことをひとがやれるならそのほうが快適だが、人手不足の解消には将来的につながることを感じたから。(どちらともいえない)
 - スピードの速い実験で見ると実感できるかも (どちらともいえない)

アンケート結果 主なコメント

- 改善点

- 現状では何とも言えないが、実装される現場での問題が違うのでその改善、修正のスピード
(実験で発生する問題に対する改修速度が重要になってくる)
- 異機種作業中、待機になるのが、複数の作業を行う場合は待機せず、別の作業を行ったりできると効率的かなと思います
- ロボットの走行の音、音がキュルキュルとなっており、ちょっと不愉快であった
- ロボットがどう動くのかわからないので、様子を見ながら通行しないとぶつかりそう。こういったルートで通るのか見せてくれるとうれしい
- ルックスが子供にウケるようになると様々な場所で人気を得られそう
- かわいくない、外見をよくしてほしい
- 配送の速度はもっと上げたほうが良いと思う

アンケート結果 主なコメント

- 意見要望

- **ぜひまた大学で実証してもらい、大学の業務に生かせるものがあればうれしいです**
- 素晴らしい実証実験だと思います。近い将来、東京駅などでも実験していただき確度を上げたシステム、ロボット制御を作り上げてほしいなと思います
- **実践に向けたシナリオをもう少し拡充してほしい**
- スーパーで売り場がわからない品物を持ってきてくれたら便利
- 去年のQRコード位置情報から、さらに発展して空間の情報を地図化するなど、比較的進化している

