OPIL: Optimizing Mobile Robot Deployment using IoT

Juan Manuel Jauregui Becker
Mobile World Capital Barcelona
EU SMEs need to sustain competitiveness

Improving productivity while keeping up with complexity growth

Enabling flexibility while being resilient to change
Implementing new technologies

Improving Competitiveness

Improving Operational Excellence
- Continuous Improvement
- Total Productive Maintenance
- Quality Management
- …

Implementing new technologies
- ERP, MES, WMS
- Robots
- End effector etc
- AVGs, Mobile Robots

Logistics account for up to 50% of total manufacturing costs

Are hitting a plateau

Smart logistics for manufacturing
SMEs & Automated Logistics Challenges

Challenge 1: Changes in production
• Uncertainty regarding future demand requirements
• Uncertainty in future product portfolio
• Layout changes specially during growing phase

Challenge 2: Available solutions on the market
• Traditionally have low adaptability
• Software integration costs are high
• System expansion is expensive & time consuming
• Vendor lock

Not interesting from production perspective!
Not interesting from economic perspective!
What SME’s do need

- Fast integration
- Low integration costs
- Easy to integrate with hardware
- Easy to expand

OPIL: IoT Platform for Logistics Automation

Developed by members of the L4MS Consortium
Factoy Equipment & Devices

3D Simulation of the factory floor

Enterprise Applications For inventory

OPIL

Sensing & Perception

Human Machine Interaction

Task planner

Suit of Software Solutions

ERP

MES

PLM

Smart logistics for manufacturign
Using OPIL and Digital Twin

**Execution phase**
Connect devices and steer logistic operations

**Planning phase**
3D factory simulation to design layout, routes, schedules

*Continuous adaptation & reconfiguration of logistic system with lower effort*
Advantages & Benefits

**Adaptation of new layouts**
Design, redesign and optimize
Speed & precision with Visual Components simulations

**Adaptation to changing production volumes**
Easy to include new AGVs, Human Agents, etc.
Changes in Production Planning & Scheduling automatically considered

**Investment estimation**
Time and installation costs drastically reduced
Time for system adaptation drastically lower
Standard interfaces

**Adaptation to new technologies**
Plug and Play functionality to sensors, ROS based Automated Guided Vehicles (AGV)
APPLY at l4ms.eu!

Funding up to 250,000€ to develop flexible, responsive logistics system

Deadline 30th November 13:00 CET

Interested but don’t have a partner?
https://l4ms-registration.fundingbox.com/

Do you have questions?
helpdesk@l4ms.eu

Send your proposal using this link!
https://www.l4ms.eu/
Thank you!

Twitter: @L4MS_Eu
Linkedin: L4MS
Slideshare: slideshare.net/L4MS
Youtube: L4 MS

http://l4ms.eu
OPIL Technical Tools

• ROS (Robotic Operation System)
  • Provides libraries specifically developed for robot control

• FIWARE Orion Context Broker (OCB):
  • Data distributed service
  • Every node of OPIL can write and read messages
  • The tool used by the nodes of the architecture to communicate
OPIL modules
Architecture

- **Software System Layer**
  - Logistics functionality
  - Simulation

- **Cyber Physical System Layer**
  - Middleware
  - Enables communication among different systems

- **IoT nodes**
  - Production floor
  - The ones that perform logistics tasks
Goal of task planner?

• Robot Fleet Management
  
  • High level fleet tasking
  
  • Optimization of logistic tasks based on specifications
  
  • Coordination of logistics workers robots/humans
  
  • Supervision of task execution
Task Planner sub-modules

• Consists of 3 different sub-modules:

1) **Task Supervisor**
   monitors the execution of the task dispatched to the agents

2) **Business Process Optimization**
   decides and optimizes the tasks to be dispatched to the different agents

3) **Motion Task Planning**
   plans the motion tasks for the robot agents
Task Supervisor

1) Receives a “Task Specification”
   • “Task Specification”: A specification of the sought logistic task in the “Logistics Task Specification Language”

   • “Logistics Task Specification Language”: A programming language introduced in OPIL that is appropriate for logistics operations

2) Parses the “Task Specification” to generate the “Task Plan” for the BPO

3) Monitors the execution of the “sequence of operations” received from BPO
   • If the execution fails it informs the BPO to provide an alternative sequence
Task Supervisor components

• **TaskScheduler:**
  - Responsible for creating and monitoring TaskManager.
  - Denotes a collection of one or more TaskManager.

• **TaskManager:**
  - Responsible for creating and monitoring Tasks.
  - TaskManager has collection of one or more Tasks.

• **Task:**
  - Describes a transport order that is being executed.
  - Has information on what will be transported, from where to which destination, how this task will be triggered.
Business Process Optimization

- Determines which robots/humans where pick up items and where to drop them off
- Handles on-the-fly optimization of tasks received from Task Supervisor
- User provides specifications of logistics tasks:
  - Resources to which machines, drop-off places and pick-up places.
- Minimizes logistics resources required for a given task.
  - Battery life, distance, types of AGVs.
Business Process Optimization

Set-up through the Human-Machine-Interface in two levels:

- SI have to first populate the system with predetermined tasks and resources in the system.
- Factory managers can, during manufacturing operations, select which tasks have to be fulfilled when.
- ERPs, or WMS can also steer the selection of predefined tasks.
Business Process Optimization

1) Receives a “Task Plan” from the Task Supervisor
   • “Task Plan”: A description of the sought logistic state vs the current logistic state

2) Calculates the required resources to fulfill the “Task Plan” by receiving input from the MTP

3) Calculates the “sequence of operations” that are required to fulfill the “Task Plan”

4) Optimizes 2. and 3.
Motion Task Planning

- Receives start and end destinations of **Robot/ Human Agent Nodes**.
- Computes best, shortest and/or fastest path for navigation
- Handles communication with **RAN** (MOD.SW.RAN).
- Its aware of states (current pose, position, current task)
- Cost is used by **the BMO** to find local optimum for input scenarios.
- Provides deadlock-free, (Near) optimal optimal path
- Avoid loops and collisions
OPIL: Human Machine Interface
Functionality

Function modules
- Determine operations
- Have SMART behavior

- Task Planner - TP
- Sensing & Perception - SP
- Human Machine Interaction - HMI

Nodes
- Perform operations
- Follow tasks

- Robot Agent Node - RAN
- Sensor Agent Node - SAN
- Human Agent Node - HAN

Run on Visual Components
- VC Robot Agent Node – VC RAN
- VC Sensor Agent Node – VC SAN
- VC Sensor Actuator Agent Node – VC A SAN

Run on OPIL Server

Run on local device
### Functionality

**Function modules**
- Determine operations
- Have SMART behavior

**Task Planner - TP**
**Sensing & Perception - SP**
**Human Machine Interaction - HMI**

**Nodes**
- Perform operations
- Follow tasks

- **Robot Agent Node - RAN**
- **Sensor Agent Node - SAN**
- **Human Agent Node - HAN**

**Run on Visual Components**
- VC Robot Agent Node – VC RAN
- VC Sensor Agent Node – VC SAN
- VC Sensor Actuator Agent Node – VC A SAN

**Run on OPIL Server**

**Run on local device**
Human Agent Node (HAN)

- Local Human Machine Interface for logistic workers
- Receive tasks
  - New tasks from Task Planner
- Read information of the tasks
  - What is the task, where to perform the task
- Send status updates
  - Task started/finished
  - Availability
  - Location information

Tablets and similar mobile devices
Workstations (Desktop, laptops)
Human Agent Node (HAN)

- Each human agent workstation has own HAN
- Unlimited number of HAN
- Via OCB from task planner:
  - Received tasks
  - Send Status updates

Orion Context Broker

- Task Planner
  - Set new task
  - Receive status updates

Human Agent 1
- Receive task update
- Update status
- Receive task update
Functionality

Function modules
- Determine operations
- Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Run on OPIL Server

Run on local device

Run on Visual Components

Nodes
- Perform operations
- Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN

VC Robot Agent Node – VC RAN
VC Sensor Agent Node – VC SAN
VC Sensor Actuator Agent Node – VC A SAN
RAN

• RAN Has two main components:
  • Core:
    • Task management:
      • In charge on managing sequence of tasks to perform
      • Allows for task updates (modifying tasks already assigned)
      • It load and unload through roller conveyors or forklifts
    • Robot navigation: In charge of managing motion according to coordinates

• RAN-AVG Interface:
  • In charge of translating motion to AGV’s native controller.
RAN Module in OPIL
**task_management_channel:** on this topic CancelTask messages are sent by the TP to the RAN; these messages will be used inside the RAN.

**motion_channel:** on this topic MotionAssignment messages commanding the root movement are sent by the TP to the RAN; these messages will be processed and low-level motion control messages will be sent to the robot.

**action_channel:** on this topic ActionAssignment messages are sent by the TP to the RAN; these messages, telling the robot which action to perform, will be processed and eventually forwarded to the robot.

**action_channel_AGV:** on this topic action data are forwarded to the AGV using ActionDefinition messages.

**status_channel:** on this topic RANState messages are forwarded by the RAN from the robot to the TP.

**description_channel:** on this topic the RobotDescription messages are forwarded by the RAN from the robot to the TP.

**cmd_vel:** on this topic are published commands that control the Robot – from RAN to HW.

**status_channel_AGV:** on this topic RobotState messages are sent from the Robot/AGV to the RAN.

**description_channel_AGV:** on this topic, RobotDescriptionAGV messages are sent from the Robot/AGV to the RAN.

**error_channel:** on this topic errors from AGV or RAN are sent to the TP via ErrorRAN messages.

**error_channel_AGV:** on this topic errors from AGV or RAN are sent to the TP via ErrorRAN messages.
Sensor Agent Node

Function modules
- Determine operations
- Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Run on OPIL Server

Run on Visual Components
- VC Robot Agent Node – VC RAN
- VC Sensor Agent Node – VC SAN
- VC Sensor Actuator Agent Node – VC A SAN

Run on local device

Nodes
- Perform operations
- Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN
SAN

An Industrial IoT (Internet of Things) module responsible for:

• Receiving raw data from sensors
• Wrapping it into a designated entity format containing necessary data
• Providing other OPIL Nodes with data
• Raspberry Pi and Revolution Pi are supported
Working principle of Sensor Agent Node

• Sensors’ raw data is received by SAN
• Connection with Cyber-Physical Middlelayer is handled automatically
• Context data is provided to other modules upon update
Development

In further versions:

• Creating a link over industrial protocols for most of the PLCs
• Making SAN a server
• Creating framework entities for most used sensors
VC SAN

Function modules
• Determine operations
• Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Run on Visual Components
VC Robot Agent Node – VC RAN
VC Sensor Agent Node – VC SAN
VC Sensor Actuator Agent Node – VC A SAN

Run on OPIIL Server

Run on local device

Nodes
• Perform operations
• Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN
Structure of VC SAN

• Mimic behaviors of real sensors
• Supports:
  • Creating entities when simulation is started
  • Update entities during simulation
  • Deleting entity after simulation
• OCB has only 1 entity per simulation
How it works

• Gathers data from sensors through component interface and translates component information to JSON format and updates the entity in OCB
• Creates, updates and deletes entity automatically
• Currently updates entity immediately after sensor data is received
  • Later it can be possible to send gather more data and send it periodically.
VC SAN Actuator

Function modules
- Determine operations
- Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Nodes
- Perform operations
- Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN

Run on Visual Components
VC Robot Agent Node – VC RAN
VC Sensor Agent Node – VC SAN
VC Sensor Actuator Agent Node – VC A SAN

Run on local device
Run on OPIL Server
VC SAN Actuator

• SAN actuator is a component which enables sending real word sensors

• SAN communicates directly with Orion CB and translate them to Boolean, real or string signals

• OCB has only one entity per SAN
How it works

• SAN actuator communication settings are similar to SAN component

• Requires to set up IP address of the OCB and valid port

• After configuring Test connection
VC RAN

Function modules
- Determine operations
- Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Nodes
- Perform operations
- Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN

VC Robot Agent Node – VC RAN
VC Sensor Agent Node – VC SAN
VC Sensor Actuator Agent Node – VC A SAN

Run on Visual Components
Run on OPIL Server
Run on local device
VC RAN Description

- VC-RAN is a component that mirrors behavior of the actual RAN
- The component enables retrieving RAN messages from OCB to VC
- Operation:
  - RAN sets navigation goals, actions and other tasks for AGV -> OCB (stored as RAN entity)
  - VC RAN reads three attributes
  - VC RAN outputs these values
Task Supervisor

- Monitors information on ongoing tasks and general status.

- Responsible for publishing information about the running tasks as well as those that were stopped (not implemented yet).

- Receives task specification from the Advanced HMI Module using formal language and parameterized task specification.

- Sends current state information to the Advanced HMI
Sensing and Perception

Function modules
• Determine operations
• Have SMART behavior

Task Planner - TP
Sensing & Perception - SP
Human Machine Interaction - HMI

Nodes
• Perform operations
• Follow tasks

Robot Agent Node - RAN
Sensor Agent Node - SAN
Human Agent Node - HAN

VC Robot Agent Node – VC RAN
VC Sensor Agent Node – VC SAN
VC Sensor Actuator Agent Node – VC A SAN

Run on Visual Components
Run on OPIL Server
Run on local device
Sensing & Perception

- Localization: Determine where are the AGVS
- Topology: specify nodes and edges.
  - Nodes are used to specify where AVGs can be placed.
  - The nodes define goals of the route planning algorithm.
  - The list of goals for a given route are send to task planner.
  - Edges specify their movement directions.

- If the geometry of the factory in case not provided.
  - Update the geometry of the factory in case differences between reality and layout model.
  - It uses SLAM to generate layout (Simultaneous Location And Mapping)
Sensing and perception modules

• **Local SP:**
  • Provides position of AGV inside the built map of the environment in which the AGV is navigating
  • Updates the map with the new sensor readings.
  • Can build the map with SLAM (Simultaneous Localization And Mapping) (using laser scan data, odometry sensors (encoders, IMU), and an initial map for map building and updating the map, and for localization within the built map.
  • Each AGV has its own Local SP that takes care of localization and mapping.

• **Central SP:**
  • Creates topology map of the factory floor plan for the Task Planner (TP) and Human Machine Interface (HMI)
Sensing and perception modules
Overview

Robots:

<table>
<thead>
<tr>
<th>ID</th>
<th>RAN status</th>
<th>Pos X</th>
<th>Pos Y</th>
<th>Theta</th>
</tr>
</thead>
</table>

Sensors:

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Pos X</th>
<th>Pos Y</th>
<th>Set position</th>
</tr>
</thead>
</table>
Floor plant management
Task Management

Sent task specifications

<table>
<thead>
<tr>
<th>TaskSpec ID</th>
<th>Status</th>
<th>Message</th>
<th>Task Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Idle</td>
<td>Success</td>
<td>position = &quot;moldingArea_palletPlace&quot; end</td>
</tr>
<tr>
<td>Position moldingPath</td>
<td>type = &quot;pallet&quot; position = &quot;moldingArea_palletPlace&quot; end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position warehouse_pos1</td>
<td>type = &quot;pallet&quot; position = &quot;warehouse_destination_pos&quot; end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>task Transport_moldingPallet</td>
<td>Transport from moldingPallet to warehouse_pos1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Current tasks in the system

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Description</th>
<th>Status</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Transport_moldingPallet</td>
<td>Finished</td>
<td>2019-03-25 12:09:38:103777</td>
</tr>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Transport_moldingPallet</td>
<td>Finished</td>
<td>2019-03-25 12:09:38:103777</td>
</tr>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Transport_moldingPallet</td>
<td>Finished</td>
<td>2019-03-25 12:09:38:103777</td>
</tr>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Transport_moldingPallet</td>
<td>Finished</td>
<td>2019-03-25 12:09:38:103777</td>
</tr>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Transport_moldingPallet</td>
<td>Finished</td>
<td>2019-03-25 12:09:38:103777</td>
</tr>
</tbody>
</table>

Sent task specifications

<table>
<thead>
<tr>
<th>TaskSpec ID</th>
<th>Status</th>
<th>Message</th>
<th>Task Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>TaskSpec_zaa8610czd2z9b06bll2z</td>
<td>Idle</td>
<td>Success</td>
<td>position = &quot;moldingArea_palletPlace&quot; end</td>
</tr>
<tr>
<td>Position moldingPath</td>
<td>type = &quot;pallet&quot; position = &quot;moldingArea_palletPlace&quot; end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position warehouse_pos1</td>
<td>type = &quot;pallet&quot; position = &quot;warehouse_destination_pos&quot; end</td>
<td></td>
<td></td>
</tr>
<tr>
<td>task Transport_moldingPallet</td>
<td>Transport from moldingPallet to warehouse_pos1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
User & System Settings

Add new user

<table>
<thead>
<tr>
<th>User id</th>
<th>Role</th>
<th>Name</th>
<th>New Password</th>
<th>New Password again</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add

Existing users

<table>
<thead>
<tr>
<th>User id</th>
<th>Role</th>
<th>Name</th>
<th>New Password</th>
<th>New Password again</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Admin</td>
<td>admin</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Update | Delete

OCB Host

10.0.0.147

Please, enter the host name / ip address of the Orion Context Broker.

OCB Port

1026

Please, enter the port number of the Orion Context Broker.

NGSI Proxy Host

10.0.0.147

Please, enter the host name / ip address of the NGSI Proxy.

NGSI Proxy Port

3000

Please, enter the port number of the NGSI Proxy.

Save
Sensor data