Robotics and Computer-Integrated Manufacturing, vol. 57, pp. 213-229, Elsevier, 2019.
https://doi.org/10.1016/j.rcim.2018.11.011

Testing the Vertical and Cyber-Physical Integration of Cognitive Robots in Manufacturing

Volker Krueger[*,†] *Francesco Rovida*[†]
Bjarne Grossmann[†] *Ronald Petrick*[‡] *Matthew Crosby*[‡]
Arnaud Charzoule[§] *German Martin Garcia*[¶]
Sven Behnke[¶] *Cesar Toscano*[||] *Germano Veiga*[||]

Abstract

In recent years, cognitive robots have started to find their way into manufacturing halls. However, the full potential of these robots can only be exploited through a) an integration of the robots with the Manufacturing Execution System (MES), b) a new and simpler way of programming based on robot skills, automated task planning, and knowledge modeling, and c) enabling the robots to function in a shared human/robot workspace with the ability to handle unexpected situations. The STAMINA project has built a robotic system that meets these objectives for an automotive kitting application, which has also been tested, validated, and demonstrated in a relevant environment (TRL6). This paper describes the STAMINA robot system and the evaluation of this system on a series of realistic kitting tasks. The structure of the system, evaluation methodology, and experimental results, are presented along with the insights and experiences gained from this work.

Keywords: autonomous robot, robot skills, kitting

[*] Robotics and Semantic Systems Group, Lund University, 22100 Lund, Sweden (e-mail:volker.krueger@cs.lth.se

[†] Robotics, Vision, and Machine Intelligence (RVMI) Lab, Aalborg University Copenhagen, 2450 Copenhagen, Denmark

[‡] Department of Computer Science, Heriot-Watt University, Edinburgh, United Kingdom

[§] PSA Groupe, Sausheim, France

[¶] University of Bonn, Germany

[||] INESC TEC, Porto, Portugal

1 Introduction

The demand for more flexibility on the factory floor requires novel robotic technologies that are able to cope with a large amount of variability, compared with classical manufacturing robots. Cognitive robots that are presently under development for the shop floor are equipped with sophisticated manipulators and a large number of sensory devices, including laser scanners and 3D cameras. The sensors are meant to help the robot manage uncertainty about its physical environment. For example, robots might need to interact with humans to coordinate tasks in shared environments. Another big asset of such a cognitive robot is that the objects it needs to manipulate are not required to be in exact locations any more. For example, it is enough for the robot to know that an object of interest is on a specific pallet. With its camera the robot will then be able to find the object.

Fig. 1: The cyber-physical STAMINA cognitive robot performing an industrial kitting task in the experimental kitting zone at PSA Peugeot Citroën.

The STAMINA project[1] explored the development of such a cognitive robot for use in an industrial *kitting* application at PSA Peugeot Citroën (hereafter, PSA). Kitting (see Fig. 2) requires the robot to collect parts specified on a list called a *kitting order*, place them into a kitting box (the two white boxes on the robot in Fig. 1 are kitting boxes), and deliver the completed *kit* to the manufacturing line. In principle, kitting can be applied to any part in the warehouse. However, from a robotics perspective, the robotic gripper limits the parts that can be handled, and a

[1] This work was supported through the EU FP7 project STAMINA, Grant no. 610917.

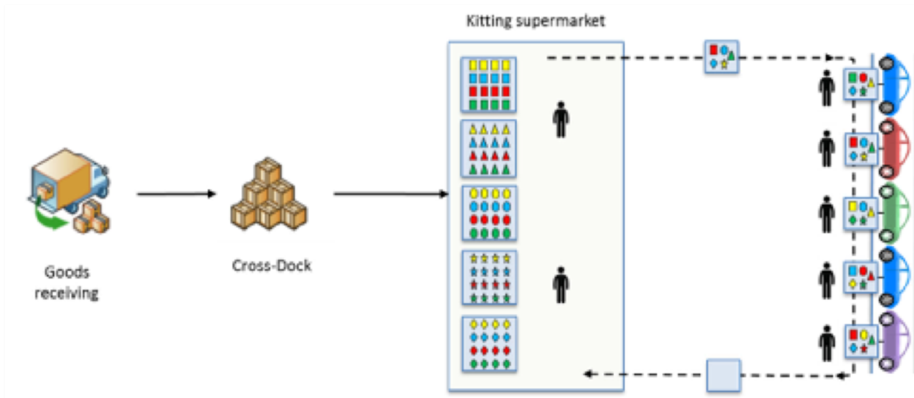

Fig. 2: An overview of the kitting process: parts are fetched from a kitting area (the kitting supermarket) and put into a kitting box that is delivered to the manufacturing line.

dedicated gripper needs to be chosen depending on the parts the robot is expected to pick. The STAMINA robot is also equipped with a special conveyor system to load and unload kitting boxes, however, kitting box logistics are not addressed in this paper.

Currently, kitting orders at PSA are printed on paper and executed by humans who collect the parts and assemble the kits. In earlier work [18], a robotics approach to this problem was presented which integrated the STAMINA robot's hardware and software systems with the *Manufacturing Execution System (MES)* at PSA, so that the robot could perform kitting tasks completely autonomously. Using this approach, kitting orders are transmitted directly from the MES to the robot via a *logistic planner*, which holds information that the robot might require for completing the kitting task (e.g., part appearances (CAD models), part quantities, warehouse layouts, and the location of boxes, shelves, and pallets). A *mission planner* interacts with the logistic planner to determine whether a kitting order is achievable by the robot, given the current information about the kitting environment. Finally, a kitting order is sent to the robot, together with the information necessary for performing the task. A robot-level *task planner* then identifies the actions the robot should actually perform in order to complete the kitting order. These actions are determined on the fly, with recovery mechanisms to handle unexpected situations as they arise. As a result, the STAMINA robot is able to collect the requested kits completely autonomously.

This paper builds on the previous work by making two key contributions. First, it describes the final implementation of the STAMINA robot used for kitting, which includes several key enabling technologies:

- The physical robot has a set of generic *robot skills* [27, 32] that allow it to perform various tasks, including those required for kitting. These include skills to *drive* [30, 31] the robot to a specific location, *pick* [14, 13] a part from a storage container, and *place* [9, 10, 11, 28, 29] it into the kitting box. The skills make use of the various cameras and laser scanners to cope with uncertainty in the shop floor environment.

- In addition to generic robot skills, the robot is equipped with dedicated skills that enable a kitting technician to *train* the system, for instance by driving the robot around the kitting area to define the layout, or teaching the robot how to pick particular objects. This trained information is stored and shared with the various components of the system as needed [18].

- The robot is able to complete a kitting order by automatically *planning* the correct sequence of skills [7, 8, 26, 34]. To do this, the robot interacts with a *logistic planner* that provides information about the available parts, their appearance, how they should be grasped and placed, and where they are located, including a map of the kitting area [18]. The result is a form of *automatic programming* that autonomously sequences skills to achieve the given kitting order.

- A suitable *human-machine interface (HMI)* was developed, together with a process for inputting the required knowledge base that describes the physical environment, following ISO standards [15, 16]. As a result, the classic robot programming task was essentially replaced by the task of building the knowledge base correctly.

- Several sources of errors were identified and automatically managed by the STAMINA robot. These included situations where the shop floor environment was dynamic, the workspace was shared with humans, deviations between the knowledge base and reality were likely, or skill execution errors were common (e.g., an object slipped out of the gripper). After detecting and identifying relevant error situations, the robot attempted to recover from errors autonomously, or otherwise called a human for help.

Second, in collaboration with PSA, the STAMINA robot was tested in a realistic setting consisting of a large factory hall with shelves, storage containers, car parts, and real factory communication protocols, in accordance with EU TRL6 guidelines [36] and the safety regulations at PSA [18]. This testing aimed to

- verify that the robot and its supporting systems were operating correctly and as expected, and evaluate the used approach [18].
- validate that the resulting system could successfully complete kitting orders communicated through the factory's existing systems to the satisfaction of a kitting technician: Are assumptions correct? Does the logistic planner hold all the needed information?

Therefore, this paper additionally:

- summarizes the design of the experimental testing, including a brief discussion of the analysis of the existing kitting processes at PSA and the formulation of *storyboards* that mimic typical situations that can arise in this environment;
- provides an account of the testing methodology used to evaluate the STAMINA robot system, including the requirements for the experimental tests; and
- describes and analyses the project's experimental results. In all experiments reported here, traveling speed of the robot and moving speed of the robot arm were constrained according to a risk analysis done at PSA before the experiments.

With the clear definition of the experimental setup, precisely described test scenarios, and a thorough evaluation of the results, this paper provides a baseline for similar future applications.

The remainder of the paper is organized as follows. In Sect. 2, related work is discussed. In Sect. 3, an overview of the STAMINA robot system is presented, along with a description of how it is integrated with the manufacturing systems at PSA, how robot skills are modelled, how the robot stores information, and how the robot plans its actions. In Sect. 4, the experimental setup is described. In Sect. 5, the results of experimental testing are presented and analysed. Finally, in Sect. 6 the overall results are discussed together with lessons learned and a summary of the STAMINA project's conclusions.

2 Relation to Other Work

This work aims to connect existing logistic approaches with advanced robotics, which aligns with recent trends in the area towards more intelligent manufacturing and factory automation. For example, [17] discusses a service-oriented architecture (SOA) for dynamically scheduling mobile robots for material supply at manufacturing lines. At a high level, there are similarities to the STAMINA architecture and approach, however, in terms of robotics, [17] focuses on container logistics. In STAMINA, robots are instead expected to handle individual parts instead of containers, which is much more challenging for two reasons:

1. Object handling is non-trivial in the settings considered here because the robots need to manage uncertainty in the environment and in the sensor data.

2. A robot that is expected to handle individual parts requires additional information such as part appearance, part location or grasping rules, and these types of information are not readily available for robot control in state-of-art logistic automation systems. One goal of the STAMINA work is to identify what additional information would be needed when using advanced robotics for kitting, including an appropriate approach for managing this data.

Kousi *et.al* [17] correctly point out the low acceptance of advanced robotic systems such as the one proposed here in industrial environments. One reason for this is the limited performance of these advanced systems which will require a considerable effort from the robotics community before improvements can be made. One aspect of the present work therefore focuses on encapsulating the low-level robotics capabilities into separate components, i.e., the skills. This way, the robotic challenges become decoupled from the high-level problems of handling and coordination. At this point, it is possible to start improving the high-level approaches, for instance using techniques from [5, 19, 22, 4, 1], apart from improving individual skills. There have also been recent efforts aimed at extending certain high-level approaches (such as automated planning) to these tasks [24]. At the same time, it is worth investigating the applicability of this approach for other use cases, e.g., for engine assembly [1] where the high-level approaches can possible be reused and where only the individual robot skills require adaptation.

3 Overview of the STAMINA Robot System

In this section, an overview of the STAMINA robot system is presented, highlighting the main components of the system along with the connections between these components, as shown in Fig. 3: the robot skills framework, the logistic planner, and the mission and task planners. These components implement a core idea in the STAMINA system, namely the distinction between *skills*, *tasks*, and *missions*: a skill is a basic competence of the robot, such as picking a part or driving to a location; skills can be concatenated autonomously into larger programs, called tasks; and missions are directives to form complete kits, according to a list of kitting orders. Together, these components will be relevant for understanding how

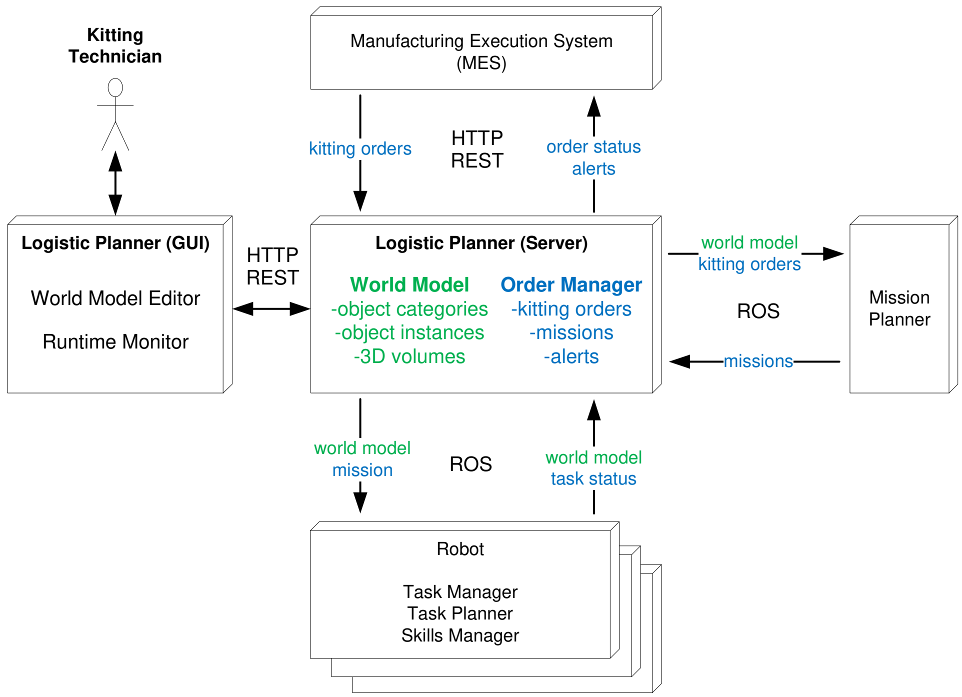

Fig. 3: STAMINA architecture: main components and information flow.

the STAMINA robot system integrates with the factory's Manufacturing Execution System, and for interpreting the experimental results in Sect. 4 and Sect. 5.

3.1 Skill-Based Robot Control

To control the robotic system, high-level robot behaviors called *skills* are used [18, 27, 35] to encapsulate the basic functionalities of the robot. Skills are meant to be self-contained and similar to apps on a smartphone. For instance, Fig. 6 shows a sequence of skills used by the robot to pick a part from a pallet and place it into a kitting box [12, 13, 14]. Skills have access to the robot hardware and other infrastructures via a software control platform called *SkiROS* [35, 32]. SkiROS is a modular and scalable Service-Oriented Architecture (SOA) for skills that runs on top of the Robot Operating System (ROS). While ROS provides a low-level interface to the robot platform, SkiROS implements the SOA on a higher level, providing an infrastructure for coordinating robot actuation, sensing, and communication.

Skills have a specific structure (see Fig. 4) and are meant to make deterministic changes when applied by the robot. To track the predicted effects of skills, SkiROS

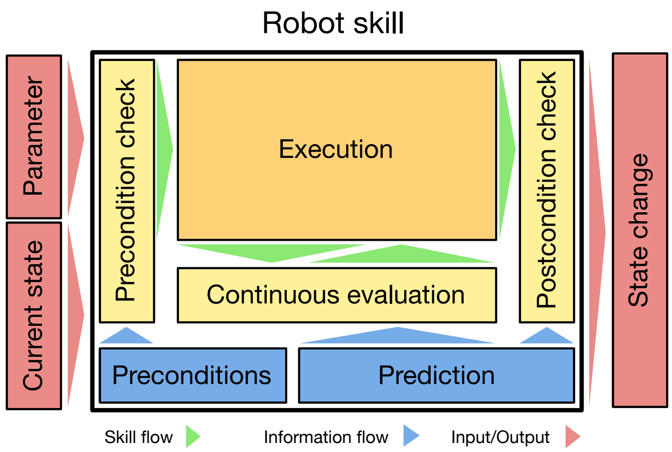

Fig. 4: The conceptual model of a robot skill. The preconditions and prediction (blue blocks) are informative aspects of the skill. The checking procedures (yellow) verify the informative aspects before, during, and after execution (orange). Execution is based on the input parameter and the robot world model (red), and results in a change to the robot world model (red).

maintains a *robot world model*. A skill's *preconditions* specify which parameters in the robot world model are relevant, and the values these parameters must have, so that the skill can be expected to execute successfully. Precondition checking follows a two-step procedure: it uses the available sensory devices (Fig. 5) to update the relevant parameters in the world state, and it verifies that these parameters have the required values. The *prediction* (postconditions) describes the changes to the robot world model resulting from a successful execution. Postcondition checking also involves two steps, as above: the available sensory devices update the relevant parameters in the world model, and the relevant parameters are checked for the required values. A skill's preconditions and prediction provide a link to the planning components of the STAMINA system (see Sect. 3.4) which automatically sequence skills into tasks.

Several skills shared some basic but important functionalities such as locating an object or computing a suitable arm trajectory. These re-occurring subfunctions of skills are called *skill primitives*. Unlike skills, skill primitives do not have a specific structure but are instead a collection of useful functions. The skill primitives used are `locate`, `arm_motion`, `kittingbox_registration`, `gripper_oc` and `ff_planner`. As their performance impacts skill performance, they will be tested in detail in Sect. 5.7.

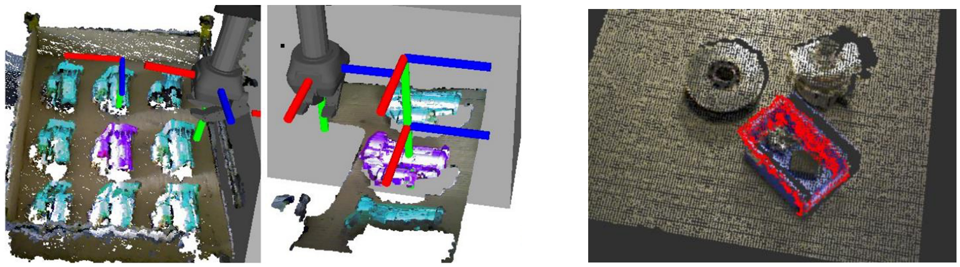

Fig. 5: The use of the sensory devices is shown for object localization of a pick skill (left), and the localization of small load carriers (SLCs, red in the right image) for a place skill (right). The techniques used here match point-clouds from RGBD cameras with CAD models of the objects [10].

3.2 Logistic Planner

The *logistic planner* is responsible for providing an interface to the MES and communicating with the rest of the STAMINA system, by a) building a knowledge base of the kitting area called the *logistic world model* which describes all of the implanted physical objects (e.g., shelves, small and large boxes/pallets, and parts), b) interacting with the MES to receive kitting orders (a list of the parts to be collected and placed into a kitting box) and returning its execution status, and c) receiving robot assignments (missions) from the mission planner (see Sect. 3.4) to be communicated to the STAMINA robot together with the logistic world model for execution by the robot. The logistic planner is not a planner in the robotics/AI sense but rather a coordinator or dispatcher, that distributes requests from the MES to the robot, along with the relevant knowledge needed for completing missions. The logistic planner coordinates between the specific functionality of the MES (responsible for scheduling the production), the mission planner (responsible for assigning missions to the robot), and the STAMINA robot (responsible for performing the kitting tasks). In addition, the logistic planner acts as a communication hub by handling different communication protocols: HTTP REST on the MES side and ROS on the robot side.

A dedicated shop floor worker called a *kitting technician* is responsible for building the logistic world model using the logistic planner's graphical user interface. In the first step, a 2D map of the kitting area is retrieved from the STAMINA robot in the form of an occupancy grid created by SLAM (Fig. 14). The kitting technician then populates the map with instances of 3D models corresponding to the objects physically available in the kitting area (Fig. 15, top), i.e., shelves containing small boxes with the kitting parts, and large boxes placed on the floor,

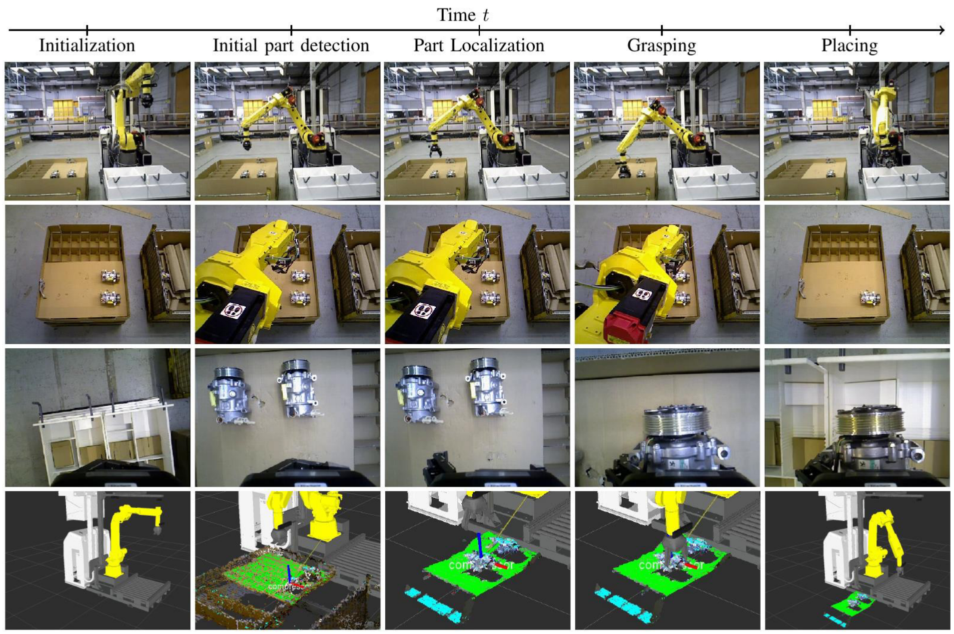

Fig. 6: Picking a part from a pallet and placing it in a kitting box. See [13] for a discussion of the picking skill.

containing layers of kitting parts (air compressors, starters, alternators). The result of this implantation process is a 3D model describing the layout of the area, the location of each shelf and large box, and their internal structure, contents and geometry (Fig. 15, bottom). It is also worth noting that the *approximate* metric 3D model that comes out of this modeling process has been proven to be precise enough for the robot.[2]. A full discussion about the reliability and real-time detection of inconsistencies in the logistic world model by the robot is provided in [2].

The logistic planner is implemented as an independent and autonomous software system, comprising two elements: a server and a user interface. The server makes use of a document-oriented NoSQL database (MongoDB),[3] assuring the persistence of the logistic world model and the current state of execution of kitting orders and tasks. The server runs on top of a Java Virtual Machine (JVM) on a Linux operating system. The user interface runs on any Internet browser and follows the HTML 5 specification. It is programmed in HTML and JavaScript,

[2] The required precision is constrained by the overview cameras that are used to find the objects. A pallet location may be $\approx \pm 25$ cm

[3] https://www.mongodb.com/

making use of the WebGL[4] standard for the 2D and 3D visualizations of the logistic world model. Details of the software implementation are provided in [37].

3.3 Robot World Model versus Logistic World Model

An important idea in the STAMINA system is that it distinguishes between two different world models: the robot world model (from SkiROS) and the logistic world model (from the logistic planner). Both world models are knowledge bases that store information about the state of the world. The robot world model includes the robot state as well as the task-relevant extrinsic information [18]. The core part of the robot's knowledge is organized into an ontology, defined in the Web Ontology Language (OWL) [3]. The ontology can be easily embedded, edited, and extracted from the SkiROS system [32]. In STAMINA, this includes information about the objects, kitting boxes, pallets, and their locations, visual appearances and grasping poses. The database also integrates with other low-level databases to include information such as the shop floor map for driving and the robot kinematic model. The extrinsic information is provided to the robot by the logistic world model. While the logistic world model holds the *entire* knowledge base of the factory hall, the robot world model holds only the task-relevant part of the logistic world model. For instance, the robot world model may only have models of a shelf and a pallet when the task involves specific instances of those containers. The many other objects in the logistic world model are not needed by the robotic system in the scope of this task.

3.4 Mission and Task Planning

The *mission planner* is responsible for assigning kitting orders to the STAMINA robot, thus creating robot missions. Kitting orders and world model information from the logistic planner are supplied to the mission planner, which checks if the kitting order is in principle achievable (e.g., there are enough parts available) and helps to identify the information the robot requires to complete the task. To do this, the mission planner uses an automated AI planner called the Agent Decomposition Planner (ADP) [6, 7] which works with a model of the robot's capabilities (also available from the logistic planner) and a problem description generated from the specific kitting order information [7]. Once a mission assignment is complete, it is communicated back to the logistic planner.

When the logistic planner receives a mission, it is communicated to the robot which then passes it to its *task planner*. The task planner interprets the mission as a request to construct a sequence of robot skills (a task) that achieves the mission's

[4] https://www.khronos.org/webgl/

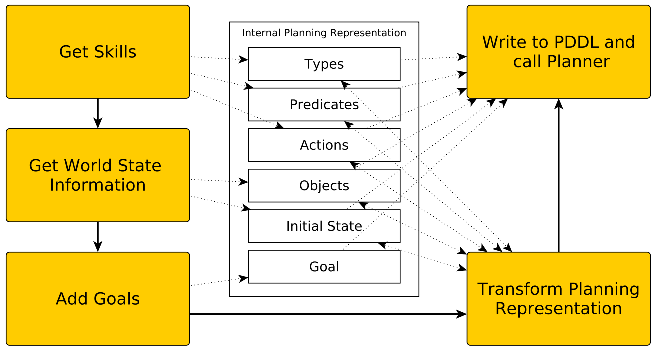

Fig. 7: Overview of the task planning process. A planning model is created in PDDL from the robot skills, world model, and mission information. The planning model is used by the planner to generate a task (a sequence of skills) that, when executed by the robot, will complete a kitting order.

corresponding kitting order. The task planner has three main functions: 1) it creates a *planning model* of the robot skills, current world model, and mission, 2) it calls an external planner that uses the planning model to construct a sequence of skills for the given mission, and 3) if successful, it returns the sequence of skills to SkiROS for execution. The planning model is generated automatically from the robot world model and skills definitions provided by SkiROS, and is represented in the Planning Domain Definition Language (PDDL) [21]–the standard language of most modern planners–which means the output is suitable for use with almost all recent planning systems. The task planning process is illustrated in Fig. 7. An example of a skill used by the task planner, written in PDDL, is given in Fig. 8. Fig. 9 shows the output of the task planning process for a mission to collect two parts; the resulting sequence of skills involves driving between locations (the `drive` skill), picking objects (the `pick` skill), and placing them in the kitting box (the `place` skill).

One additional feature of the mission and task planning components is that they can be used to automatically recover from errors, for instance by replanning skill sequences that fail during execution. For example, consider the situation where the logistic planner receives a kitting order that involves picking a particular object. The logistic planner sends a request to the mission planner to check if this mission is possible. The mission planner checks a) that the object is available in the warehouse, b) where the pallet, container, or shelf that held the object is located, and c) if a robot could reach that location. However, the mission planner generally does not know where the objects are exactly located on the pallet or shelf, or whether the path to the object is blocked. These details are instead discovered by the robot

```
(:action drive
    :parameters
        (?finalLoc - Location
         ?preLoc - Location)
    :preconds
        (and (can_drive ?preLoc ?finalLoc)
             (RobotAtLocation ?preLoc))
    :effects
        (and (not (RobotAtLocation ?preLoc))
             (RobotAtLocation ?finalLoc)))
```

Fig. 8: The `drive` skill used by the task planner, modelled in PDDL.

```
drive loc-1 lbox-10
pick  lbox-10 gripper-6 t_shield
drive lbox-10 lbox-9
place grip-6 t_shield celld-19 kit-15
pick  lbox-9 gripper-6 starter
place grip-6 starter cellb-17 kit-15
```

Fig. 9: A sequence of skills constructed by the task planner in simulation.

itself with the help of its skills, sensory devices, and task planner. When the robot receives the mission, the task planner constructs a sequence of skills (e.g., involving `drive` and `pick` as in Fig. 9). However, if during the execution of a `drive` skill the robot detects with its laser scans that the planned path is blocked, a failure message would be sent to the task planner which will attempt to find an alternative route by replanning. While this example is quite trivial, in the case of a more complex kitting order, the task planner could be directed to identify the most efficient route to pick all objects, responding to unexpected situations on the fly.

The mission and task planning systems have also been tested in simulation for a fleet of robots, where robots with potentially different capabilities (i.e., different skills) can each carry out multiple kitting orders [7].

4 Experiment Preparations

4.1 Description of the Test Sprint Environment

In order to enable realistic experimentation of the STAMINA robot, a replica of a small 1,200 m^2 logistics kitting zone was created at PSA with containers and shelves for testing the STAMINA robot and its supporting system. Fig. 10, shows four aisles with a number of pallets and shelves (left) and a photo of the real kitting area with safety fences (right). Fig. 11 shows an RViz image of the robot in the kitting area (top) and an example of the shelves with larger boxes holding engine pipes (bottom).

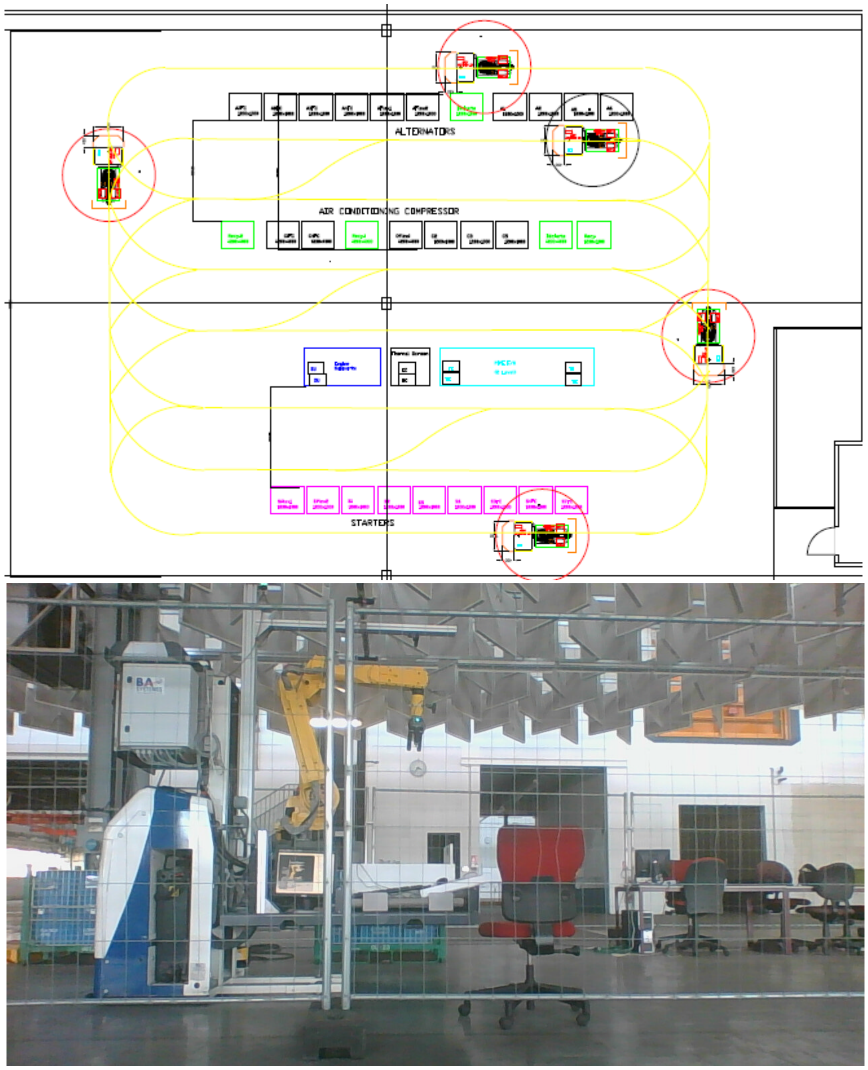

Fig. 10: Layout of the kitting supermarket (top) and a photo of the real kitting area with safety fences (bottom).

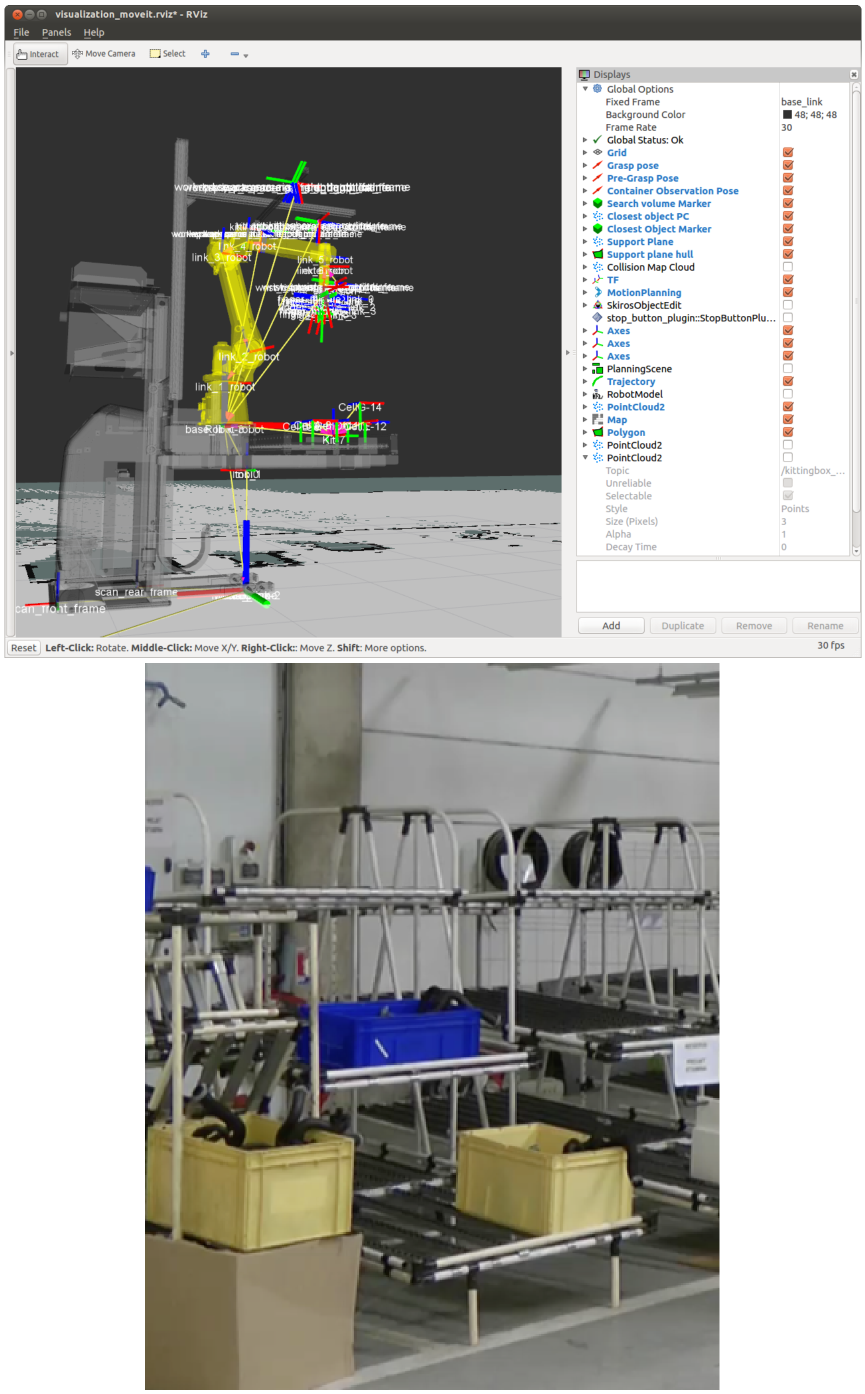

Fig. 11: RViz virtual simulation of the robot and parts of the kitting area (top), and example shelves with boxes holding engine pipes (bottom).

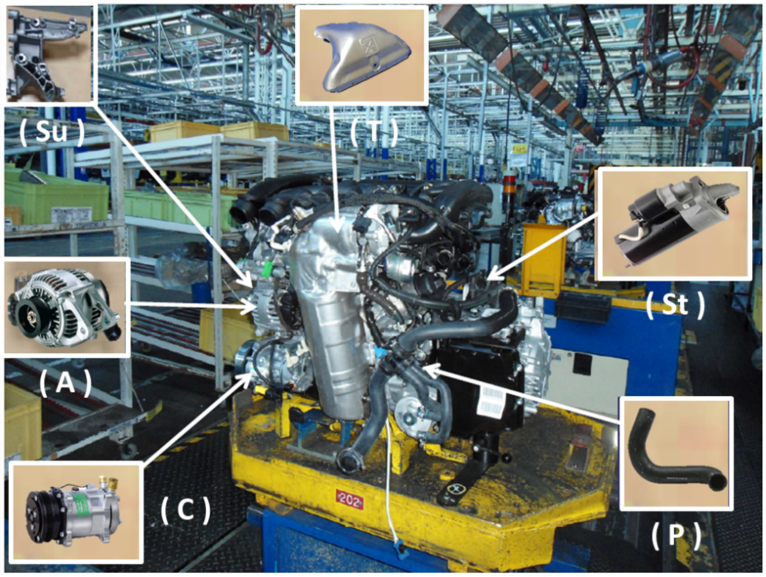

Fig. 12: Selected parts for kitting in their final engine configuration.

The testing area also included a selection of engine kit parts: starters (St), alternators (A), air conditioning compressors (C), engine pipes (P), thermal shields (T), and engine supports (Su). These parts are shown in Fig. 12 in their final assembly configuration. Furthermore, multiple types of starters, alternators, and compressors were also available and treated as distinct parts, with different visual appearances and different locations in the kitting area. This variability enabled a variety of engine kits and kitting orders to be tested.

4.2 Storyboards used during the Test Sprint

To support realistic testing, a set of *storyboards* was developed and used to guide the experimental sessions. The storyboards described realistic situations on the shop floor that were identified through a discussion process during several meetings with PSA staff:

Test 1: Setup of the logistic world model using the HMI

This test involved initializing the logistic world model using the logistic planner's human-machine interface, and was carried out by kitting technicians from PSA. The testing process involved configuring the kitting zone by specifying the locations of physical objects (e.g., shelves, small and large boxes, and parts):

(a) A 2D occupancy grid map created by SLAM is retrieved from the robot.
(b) The kitting technician populates the map with representations of the containers in the area: shelves, small and large boxes.
(c) The kitting technician specifies which type of parts are contained in which containers.

Test 2: Normal operation of the STAMINA robot

This task tested the normal (ideal) day-to-day operation of the STAMINA robot in the shop floor environment:

(a) The robot receives a kitting order from the logistic planner.
(b) The robot generates a task and begins to collect the requested parts. The entire process runs smoothly: picking, placing and drive skills work as expected and preconditions and postconditions of the skills catch potential errors. In the case of such errors, the task planner is able to replan to complete the task.
(c) At the end, the robot delivers a correctly filled kitting box to a predefined location.

Test 3: Startup and shutdown of the STAMINA system

This task tested the startup and shutdown procedures of the STAMINA robot:

(a) At the start of its shift, the robot connects itself to the logistic planner and begins its normal operation.
(b) At the end of the shift, the robot drives to its parking location.

Test 4: Installation of a new robot into the existing STAMINA system

This task tested the process for connecting new robots to an operational STAMINA system:

(a) After switching on a new robot, the robot automatically identifies itself to the STAMINA system.
(b) The STAMINA system then automatically includes the robot in future tasks. Human intervention should ideally be unnecessary on the robot system.

Test 5: Typical operation with autonomous error handling

This task tested a number of typical operational scenarios where the robot can detect and automatically handle errors through task replanning:

5.1 Object A is on two different pallets. Pallet 1 is empty. The robot updates the world model, requests filling of Pallet 1, replans and uses Pallet 2. Next kitting task excludes Pallet 1 until Pallet 1 is refilled and the world model is updated accordingly.

5.2 Object A is on two different pallets. Pallet 1 mistakenly holds Object B. The robot requests attention, replans and uses Pallet 2.

5.3 Object A is on three different pallets. Pallet 1 mistakenly holds Object B. The robot requests attention, replans and uses Pallet 2. Pallet 2 is empty. The robot replans to use Pallet 3.

Test 6: Typical operation and error handling with human assistance

This task tested a typical operational scenario where the robot can detect an error but cannot handle it autonomously (human intervention is required):

> Object A is on two different pallets. Pallet 1 mistakenly holds Object B. The robot replans to use Pallet 2, but Pallet 2 turns out to be empty. The robot calls for human assistance.

As part of the validation process for Tests 2–6, a series of metrics was identified and data was collected to help evaluate the testing results in each case. In particular, the following information was recorded for each kitting order:

- the number of parts n in the kitting order (where, typically, $1 \leq n \leq 6$),
- the number of successfully executed kitting orders,
- the number of Technician Assistance (TA) requests when the system detected a problem it could not solve on its own,
- the average execution time per n-part kitting order.

The following measures were also recorded for each skill and skill primitive:

- the number of attempts to execute the skill/skill primitive,
- the number of successful executions: in the case of skills, satisfied postconditions implied a successful skill execution,
- the number of failures: in the case of skills, a failure is either due to unsatisfied preconditions or postconditions,
- the average execution time of each skill and skill primitive.

The goal of Tests 1–4 was to verify and validate the basic functionalities of the STAMINA robot that were found to be important to PSA engineers. The goal of Tests 5–6 was to stress test the system against typical unexpected events. While it is difficult to predict the unexpected [20], PSA engineers shared their experiences to help define these storyboards and produce realistic "unexpected" situations to test. It should also be noted that the behavior of the robot in these unexpected situations was *not* hard-coded into the system, but handled by the planner on the fly. Furthermore, Tests 5–6 were defined by PSA exclusively for the experiment sessions reported in this paper in order to minimize the possibility of the system having prior knowledge of these unexpected situations. In fact, the SkiROS and STAMINA systems were designed prior to these storyboard definitions, and during earlier experimentation the system was *only* tested against the Test 2 storyboard.

5 Experimental Results

During testing, 57 kitting orders were executed with varying numbers of parts. Based on these kitting orders, the task planner generated 57 tasks, referred to as *kitting tasks*, which resulted in the execution of 608 skills and 1,754 skill primitives on the robot. The performance and debugging statistics of all kitting tasks, skills and skill primitives were automatically recorded on the robot and stored for subsequent evaluation. All kitting tasks were executed using the available parts, including two types of starters, alternators, and air conditioning compressors. Starters, alternators and compressors come well-ordered on pallets. The engine supports were placed in Styrofoam fixtures in a box to assure that they have suitable poses for being grasped. The engine pipes were randomly stored in a box. All pallets and boxes had their own locations in the kitting area. The boxes with the engine pipes and the engine supports were stored on the lower level of the racks (see Fig. 11).

5.1 Setup of the Logistic World Model using the HMI

For the robot to perform kitting operations in the kitting zone, a model describing the structure, location and containment relationships of the physical objects must initially be built by a kitting technician using the logistic planner's HMI. The assessment scenario in Test 1 targeted the usability and utility of the logistic planner in order to reduce the technician's training phase, to ensure the acceptability of future users, and to minimize the chance of inserting erroneous information to reduce the need of subsequent corrections. This was achieved in accordance with Nielsen's acceptability model [23] which divides system acceptability into two initial dimensions of social and practical acceptability. The practical acceptability dimension was selected, with a focus on usefulness in terms of utility (i.e., does

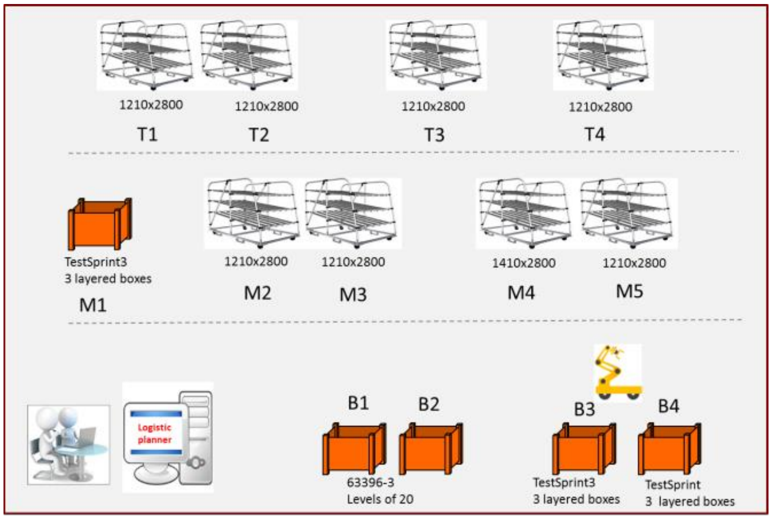

Fig. 13: Sketch of the kitting area (T means Top area, M means Middle area and B means Bottom area).

the system meet an actual need and what kind of benefit does the system provide to the worker?) and usability (i.e., how easy is the system to learn, to use, to remember, etc.).

Six volunteers from PSA participated in the HMI user testing: two kitting technicians and four *kitting monitors*.[5] Users were seated in front of the kitting area so they could look at the physical area, and they were provided with a sketch of its contents in terms of containers and parts (see Fig. 13). Each user session started with a 10 min introduction where the goals of the project and the assessment process were presented, followed by a brief look into the kiting area and a very brief introduction to the HMI. Then, each volunteer was given 30-45 min to work through a 4-step workflow which involved: 1) preparing the map, 2) placing the containers in the map, 3) associating parts to containers, and 3) checking the consistency of the world model. The volunteers were requested to complete the tasks on their own without any help. The instructor only intervened and provided support when major difficulties prevented a user from progressing in the task.

In the first step (preparing the map), the volunteers were informed that the robot's laser scanners were roughly 10 cm above the ground, that obstacles one meter above the ground could not be sensed, and that the robot's laser scanners

[5] Kitting operators execute kitting orders, while kitting monitors help, support and monitor up to five kitting operators to assure smooth kitting operations. STAMINA aims to automate the job of the kitting operator.

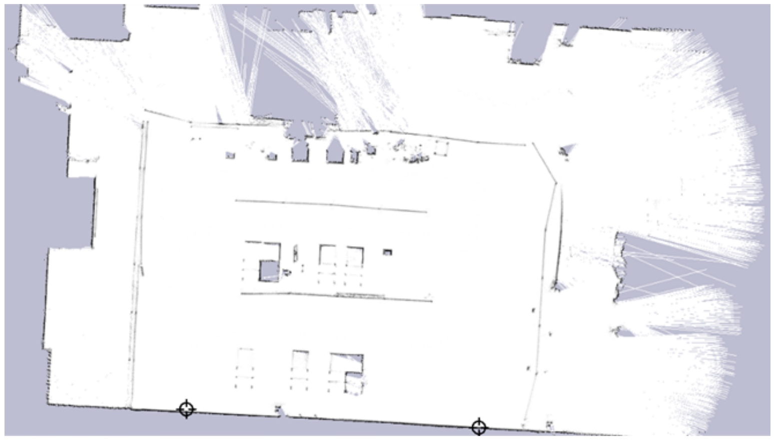

Fig. 14: Occupancy grid map of the kitting area.

created 2D measurements on a plane (see Fig. 14). Furthermore, the volunteers were provided with a paper sketch of the kitting area (see Fig. 13). By looking at the laser scan-based map on the HMI (see Fig. 14), the participants had to recognize and identify the different containers (shelves and large boxes) present in the kitting area as shown in the paper sketch. Only one participant succeeded in identifying the large boxes, and no participant was able to recognize the shelves. In addition, no participant detected that the image was not in the correct orientation. The instructor had to inform all participants that a 180° rotation was needed to align the image with the viewing direction of the participant, and the procedure needed to perform the rotation on the HMI had to be demonstrated.

The main conclusion is that data from the laser range scanners is too abstract for the shop floor workers without training. Despite being metrically and geometrically correct, the 2D occupancy grid map cannot easily be improved without the interaction of experts. To the knowledge of the authors, there are no ongoing attempts to automatically add semantic knowledge to the grid map that could aid the worker. One solution might be to supply dedicated training or to minimize the problem by adding opaque elements to the shelves so that its representation in the map is more visible. Furthermore, rotations of the map should be avoided as this type of operation is not easily seen in the 3D space by shop floor workers. However, as the container boxes are aligned in rows and have the same orientation, the scanning of the area by the robot can be made in such a way that complex orientations (greater than 45 degrees) are avoided.

Minor inefficiencies and absence of information in the HMI were detected on the following two steps of the work flow: setting the containers in the map and associating parts to containers. Subsequent development on the HMI resolved all these issues. One such inefficiency was when the kitting technician implanted an object into the map, he/she originally needed to manually update the 2D view which turned out to be inefficient. This inefficiency was improved by automating this update. Another example of how efficiency was improved was to allow the kitting technician to associate labels to abstract specification numbers of objects when populating the map, thereby making it easier for the kitting technician to remember the object specifier when consequently associating the parts to containers.

The last task was to compare and verify the 3D model with the real kitting area. This was accomplished by looking at a 3D visualization of the logistic model in the HMI (see Fig. 15, bottom). This was observed to be intuitive as the users were easily able to interact with the 3D image by enlarging and reducing the image, to visually recognize the different containers, drive around them, etc.

During the final interview, some of the participants revealed that being able to monitor the kitting robot in real time by animating the robot in the 3D visualization of the logistic model is neither realistic nor needed, and that they only required alerts if the robot faced problems. As a result, the most effective use of the 3D logistic model visualization could be seen as its ability to check the adequacy of the model.

5.2 Baseline Kitting Orders

While Sect.5.1 focused on evaluating the human machine interface using Test 1, the following tests instead focused on evaluating the STAMINA robot and the complete STAMINA system.

To test the STAMINA system, a performance baseline was generated which marked the system performance under ideal shop floor conditions. Here, SkiROS, the task and mission planners, and the logistic world model were tested, while sensing and actuation within the skills was not particularly challenged. For this baseline, the Test 2 storyboard was used: the MES requested kits with 1-6 parts. The engineers ensured perfect conditions for the robot: no obstacles were in the way, all data in the logistic world model was correct, parts were all available in the right pallets and boxes in the warehouse. For each kit, the parts were randomly selected from a set of parts that were known to be graspable by the robot with high reliability: (a) starter engine A, (b) starter engine B, (c) alternator A, (d) alternator B, and (e) air conditioning compressor. Test 2 was repeated 30 times under ideal conditions and data was collected from each trial. The results are summarized in Table 1. The robot completed its kitting task and successfully delivered the complete and correctly assembled kitting box. The column *Trials w/o replanning* shows the trials where replanning was not needed because everything worked as

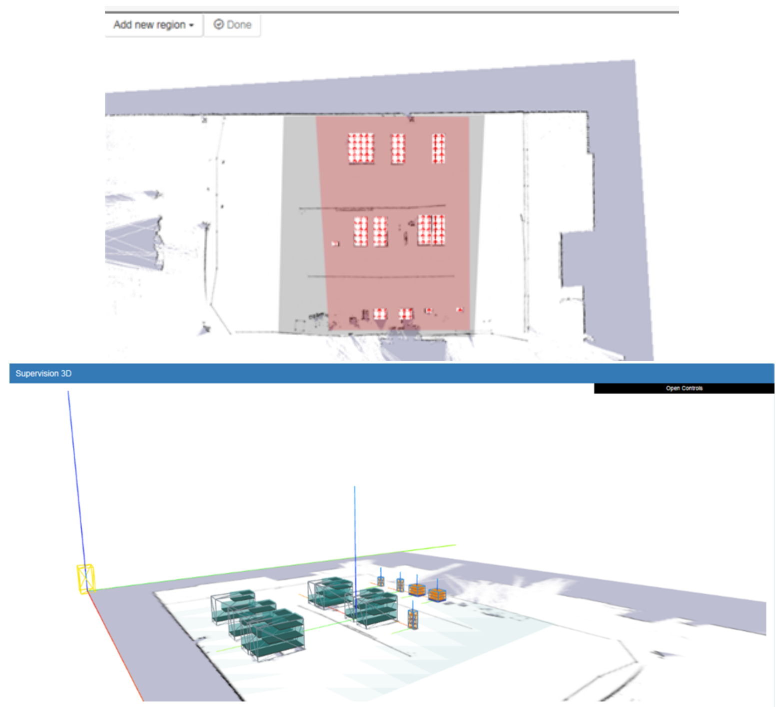

Fig. 15: Setup of the logistic world model (accessible on a web browser): 2D occupancy grid map populated with shelves and large boxes (top), and the corresponding 3D visualization (bottom).

planned. The column *Time w/o replanning* shows the average time in seconds, $\pm\ \sigma$, with σ being the standard deviation of the execution speed of the trials. The column *Trials Overall* shows the complete number of trials. These results show that even under optimal conditions, replanning was necessary for a number of reasons:

- *Robot was too far away from the goal*: The robot recovered from this problem by initiating another attempt to get closer to the goal.
- *No arm trajectory was found*: Finding a robot arm trajectory for picking was a subroutine (skill primitive, see below) of the picking skill. This problem was solved by simply restarting the picking skill, thereby taking a new image with the overview and wrist cameras.

The robot was allowed to retry a skill up to 3 times. In all cases where replanning was needed, the robot succeed on the first or second retry.

Tab. 1: Summary of the baseline tests results. The table shows that in many cases, re-planning was necessary to complete the task.

Kitting Task	Trials overall	Trials w/o replanning	Time w/o replanning	Overall time
5 parts	4	3	801s ± 51	856s ± 105
4 parts	6	3	638s ± 6	723s ± 96
3 parts	3	2	518s ± 6	593s ± 106
2 parts	3	2	316s ± 13	325s ± 16
1 part	14	10	190s ± 27	234s ± 105

5.3 Startup, Shutdown, and New Robot Installation

The ability to easily startup/shutdown the system and install new robots were explicitly identified as important requirements from the end-user perspective. The STAMINA system was designed to conform to these requirements, and thus Test 3 and Test 4 can be viewed as simple function tests of these capabilities. In particular, Test 3(a) and Test 4 simply confirmed the correct operation of the logistic planner to register robots on the STAMINA system, as discussed in Sect. 5.1.

Test 3(b) tested the parking ability of the STAMINA robot, as part of the shutdown process. Table 2 summarizes the results these experiments which were repeated ten times, with eight successes and two driving failures with the robot being more than 10 cm away from the correct parking location. The parking experiment also mimics parking for automatic charging.

Tab. 2: Summary of the parking test results. In the 2 part case, the robot was more than 10cm away from its designated parking location.

Kitting Task	Attempts	Success	TA requested	Averate time (slow)	Success Rate
Parking	10	8	0	29s	80 %

5.4 Complex Kitting Tasks

In this section, more complex kitting tasks are considered which are systematically tested using the four storyboards in Test 5 and Test 6.

Test 5.1

Scenario: *Object A is on two different pallets. Pallet 1 is empty. Robot updates the world model, requests filling of Pallet 1, replans and uses Pallet 2. During the next kitting task, Pallet 1 is not used.*

For this experiment, the scenario and the world model needed to be prepared accordingly, and the necessary updates to the world model were completed within 5 min via the logistic planner user interface. The robot was immediately able to execute this kitting task successfully. Upon detecting the lack of parts on Pallet 1, the robot informed the kitting technician (*Alert: Pallet with ID <> empty*) via the supervisory display of the logistic planner and replanned to fetch the part from Pallet 2. When a new kitting order was generated, the task planner on the robot immediately took into consideration the fact that Pallet 1 was empty. To repeat the trials, the world model had to be appropriately reset. The results are summarized in Table 3: Each row shows two attempts. On the first trial, the robot was unaware of the lack of parts, which generated the TA request. On the subsequent trial, the robot planned accordingly so that no additional TA request was issued.

Tab. 3: Summary of the Test 5.1 results. The robot alerted the kitting technician twice, i.e., the first time, the robot detected that an object was missing. Once the world model was updated, the robot directly drove to the pallet with the available part so that no additional TA request was issued.

Kitting Task	Attempts	Success	TA requested	Averate time (slow)	Success Rate
5 parts	2	2	1	928s	100 %
4 parts	2	2	1	723 s	100 %

Test 5.2

Scenario: *Object A is on two different pallets. Pallet 1 by mistake holds Object B. Robot requests attention, replans and uses Pallet 2.*

This test was also correctly executed by the robot: upon detecting a wrong part on Pallet 1, the robot sends a message to the kitting technician (*Alert: Wrong part on pallet with ID <>?*) and continues with replanning its kitting task. As in Test 5.1, the scenario was prepared and the world model was set up using the logistic planner's user interface. The results are summarized in Table 4: As before, the robot only issued the TA request the first time it discovered the error and then planned the subsequent kitting order accordingly.

Tab. 4: Summary of the Test 5.2 results. The robot alerted the kitting technician twice, i.e., the first time, the robot detected that an object was wrong. Once the world model was updated, the robot planned accordingly in the subsequent trial.

Kitting Task	Attempts	Success	TA requested	Averate time (slow)	Success Rate
5 parts	2	2	1	935s	100 %
4 parts	2	2	1	748 s	100 %

Test 5.3

Scenario: *Object A is on three different pallets. Pallet 1 by mistake holds Object B. Robot requests attention, replans and uses Pallet 2. Pallet 2 is empty. Robot replans to Pallet 3.*

This test is a mixture of the previous two test, and demonstrates the power of using a task planner and the skill-based approach. In particular, the robot is able to dynamically and automatically find a solution that is within the space of possible skill sequences. The situation in this storyboard easily occurs when a kitting technician inserts faulty data into the logistic world model. This storyboard shows that the robot is to some extent robust with respect to errors in the logistic world model.

The results are summarized in Table 5. It is interesting to compare the timings of Table 3 with those of Table 1: due to the required replanning, the execution of a kitting order takes considerably longer. Because the robot discovered two errors, it also issued two TA requests. The second 4 part mission also failed during this test.

Tab. 5: Summary of the Test 5.3 results. The robot alerted the kitting technician twice on each first time attempt. Once the world model was updated, the robot directly drove to the pallet with the available part.

Kitting Task	Attempts	Success	TA requested	Averate time (slow)	Success Rate
5 parts	2	2	2	1415 s	100 %
4 parts	2	1	2	599 s	50 %

Test 6

Scenario: *Object A is on two different pallets. Pallet 1 by mistake holds Object B. Robot replans to use Pallet 2, but Pallet 2 turns out to be empty. Robot calls for human assistance*.

This test case is similar to the previous ones with the main difference being that the robot in this case is unable to complete its kitting order: the robot first detects the wrong part and sends an alert to the kitting technician. It then replans to pick from Pallet 2 which it finds empty. After again sending an alert to the technician about Pallet 2 being empty it attempts to replan. The replanning fails because there is no other pallet that could provide the required part. The robot therefore aborts the kitting task with a request to the kitting technician for assistance (*Technician Assistance needed: No plan to complete the kitting order*). In the 4 part kitting order, the failure happened with the first object; in the 5 part kitting order, the failure happened with the second object. The results are summarized in Table 6.

It is interesting to mention that all the task planner can do is *attempt* to find a plan. If a plan is available, it is usually found within 1 s. A 2 s threshold was therefore set after which the planner reports “no plan found” and aborts planning. The reason for this is that the planner is unable to assess *why* a plan was not found, i.e., all it can do is to give a binary plan/no-plan answer.

Tab. 6: Summary of the Test 6 results. The robot failed in all cases because one of the required parts was not available. For the 4 part kitting order, the first part was missing; for the 5 part kitting order, the second part was missing.

Kitting Task	Attempts	Success	TA requested	Averate time (slow)	Success Rate
5 parts	3	0	3	371 s	0 %
4 parts	2	0	2	192 s	0 %

5.5 Discussion of the Test Results

During the baseline tests, it became apparent that in all experiments, the picking skill worked very reliably for de-palletizing compressors, alternators and starter engines. Picking the engine pipes proved to be more challenging as they are non-rigid and needed to be picked from a bin where the pipes easily intertangled. The engine supports were stored in a small load carrier (SLC), and picking became very challenging due to the confined space.

Contrary to many earlier test sprints, no situation was observed in the final experiments where the system was not in a well-defined system state. Here, a system state was considered to be not well-defined if (a) the world model of the robot deviated from physical reality, and (b) the robot was not able to detect this deviation. This happened easily when the robot mixed up two parts, or when a picked object slipped out of the gripper without the robot realizing it. In both cases, the robot should have called the kitting technician for assistance. But instead, the robot was not able to identify these situations and delivered kitting boxes with wrong or missing parts, and then reported the kitting task as successfully completed. The first situation was solved by improving the computer vision approaches in the `locate` skill primitive (see below) [9]. The latter situations were prevented by adding one additional but apparently important sensing mechanism: binary pre- and post-condition checks for "object in gripper" for the pick and the place skills (the `gripper_oc` skill primitive, see below). Based on these additional sensing capabilities, the robot was able to identify situations where kitting technician assistance was needed.

Table 7 summarizes the results of these experiments: the robot successfully called the technician 13 out of 15 times. The two cases where the robot did not inform the technician were when the robot failed to identify that its parking location was incorrect. Extensive additional testing for detecting TA situations was done and clearly confirmed that the only difficult situation in the end remained detecting the imprecise parking location.

Tab. 7: Summary of the TA request statistics. Out of 15 TA requests the robot should have made, it only made 13 TA requests. It failed to identify two wrong parking locations which should have led to two more TA requests.

Kitting Tasks	Should have requested	Correctly requested	False positives	False negatives	Success Rate
TA requests	15	13	0	2	87 %

For the above experiments, a conservative threshold was used and parameter settings for all skills were chosen to ensure the highest possible reliability. E.g., high-resolution images were used for computing object poses and more time was spent in finding good arm trajectories. However, these settings also resulted in slower execution times. Table 8 summarizes the Test 2 experiments with parameter settings for improved speed: camera images were subsampled by a factor of two before processing, and for computing the best arm trajectory, the number of trials for computing inverse kinematics solutions was reduced from 128 to only 32.

Optimizing the balance between parameter settings and execution speed is one possibility for improving skill execution. Even though the reliability of the skills has not been systematically explored with respect to different parameter settings, it became apparent that the impact of these parameter settings on the final skill execution speed was limited. A big reason for the slow execution speed was a) the limited robot speed, and b) the lack of coordination of the processing steps within each skill. Concerning a), the speed of the mobile platform and the arm had to be limited for safety reasons. Here, new strategies and regulations need to be developed to allow higher robot speeds without endangering human co-workers. Concerning b), it needs to be explained that during the experiments all processing steps, i.e., the execution of skill primitives inside the skills, were done strictly sequentially. Here, there is a potential for improving skill speed. For instance, one approach that has been outlined suggests using extended Behavior Trees to optimize skill primitive selection inside the skill [33].

The success rates in Tables 3–6 were calculated based on the ratio between the successful completion of kitting tasks versus the requested TAs, which is consistent with how a human production manager would evaluate success. From a systems engineering perspective, the SkiROS system behaved as expected throughout the experiments, and an improvement in success rates would require improvements to the individual robot skills.

Tab. 8: Average execution speed $\pm\,\sigma$ of the kitting tasks using the Test 2 setup but with improved parameters for faster execution speed.

Kitting Task	Attempts	Success	TA requested	Averate time (fast)	Success Rate
5 parts	3	3	0	641s±41	100 %
1 part	10	10	0	147s±18	100 %

5.6 Testing the Robot Skills

All kitting tasks from the previous section were executed using three different kinds of skills: the `drive` skill, the `place` skill, and the `pick` skill. All skills were extensively tested, either as part of the kitting tasks or manually. The performance statistics of the skills were automatically recorded and evaluated. Table 9 provides a summary of these statistics. The *Attempts* column shows the overall number of executions per skill. The *Post-conditions satisfied* column refers to skills being completed successfully with satisfied post-conditions. The *Pre-/post- conditions* *not satisfied* column counts how often pre- or post-conditions failed to be satisfied in the tests.

When interpreting the results of Table 9, it is important to keep in mind that the skills are executed based on a task planner that is planning and replanning kitting tasks as required. This means that if the pre- or post-conditions of a skill are not satisfied, the system replans. In most cases, this leads the robot to re-attempt skill execution, either immediately or at a later time.

In terms of the kitting tasks summarized in Tables 1–6, the reported number of skill executions does not correlate with the intuitively expected number of skill executions. There are two main reasons for this. First, in the case of completed kitting tasks, all skills eventually ended with “post-conditions satisfied”, even if intermediate replanning (and therefore additional skill execution) was necessary. Second, in cases where kitting tasks ended with “TA requested”, one skill either ended repeatedly with “condition not satisfied”, or the planner was unable to find any plan to complete the kitting task.

Tab. 9: Skill executions from experiments Test 2–Test 6.

Skill	Attempts	Post-conditions satisfied	pre-/post- conditions not satisfied	Averate time (fast)	Success Rate
drive	216	181	35	20.69s (17.14s)	84 %
pick	236	154	82	110.43s (78.68s)	65 %
place	156	141	15	34.34s (28.59s)	90 %

One could also ask how well the robot might have performed without the use of a planner. Taking the “post-conditions satisfied” rate for each skill in Table 9 as its *success probability*, it is possible to compute the probability of successfully executing an n-part kitting task for a predefined and fixed skill sequence:

$$\begin{aligned} &P(\text{successful } n\text{-part mission w/o planning}) \\ &= (\mathcal{P}(\text{drive}) \cdot \mathcal{P}(\text{pick}) \cdot \mathcal{P}(\text{place}))^n \cdot \mathcal{P}(\text{drive}). \end{aligned} \tag{1}$$

These probabilities are explicitly listed in Table 10 for different values of n.

5.7 Testing Skill Primitives

The results from testing individual skill primitives are shown in Table 11. Overall, skill primitives were tested 1,754 times, with a breakdown of individual attempts and successes for each skill primitive shown in the table. The *Operation Fails*

Tab. 10: Theoretical success rates when a robot is unable to replan. Eq. (1) is used to calculate the theoretical success rates for kitting tasks with varying numbers of parts, where skill sequences are assumed to be preprogrammed and fixed, with known success rates for each skill.

Kitting Task	Theoretical Success Rate
6 parts	1 %
5 parts	3 %
4 parts	5 %
3 parts	10 %
2 parts	20 %
1 part	41 %

(OF) column indicates a number of particular failures for the individual skill primitives. For instance, the `locate` skill primitive is used for locating an object on a pallet, bin, or SLC, as a prerequisite for picking. An OF for this skill primitive indicates a failure in localizing the object, meaning the pick skill also fails with a pre-condition failure. The skill primitive `arm_motion` computes a suitable arm trajectory for picking or placing an object. An OF for this primitive indicates the number of cases where the skill primitive failed to compute a trajectory, leading to pre-condition failures for the picking and placing skills. The skill primitive `kittingbox_registration` computes the exact location of the kitting box on the robot. An OF occurs when the primitive fails to compute the pose of the kitting box with low uncertainty, which can lead to a pre-condition failure for the placing skill. The `ff_planner` primitive is executed each time a task plan is required, and it successfully delivered a plan in 91% of the cases. However, in 21 cases `ff_planner` could not find a plan. These failures were counted as OFs. Each failure case was also cross-checked by a human expert who could not find a plan either. (E.g., in Test 6, there was in fact no plan and the robot correctly called for technician assistance.) Finally, the skill primitive `gripper_oc` is used to detect if an object is present in the gripper, which is a pre-condition for the pick and place skills. An OF for this primitive indicates a failure in the detection process. This skill primitive proved to be essential for checking the post-conditions of the pick and place skills, and the overall capability of the system to perform its kitting tasks.

For all skill primitives, OFs appear to be due to shortcomings of the skill primitives. While all skill primitives worked with very high reliability inside the research labs, the shortcomings only became apparent during testing in the real-world envi-

Tab. 11: Summary of the performance of individual skill primitives. The *average time* provides execution times for both slow and fast executions.

Skill Primitive	Attempts	Success	Operation failed	Averate time (fast)	Success Rate
locate	214	208	6	3.41s (3.26s)	97 %
arm_motion	650	639	11	11.95s (10.53s)	98 %
kittingbox_ registration	237	223	9	6.15s (4.20s)	94 %
gripper_oc	309	309	0	2.20s(2.07s)	100 %
ff_planner	244	223	21	5.72	91 %

ronment. One factor which may have contributed to the limitations of the picking skill in particular is that it was developed at a different place—a university robotics lab—with slightly different hardware: a smaller stationary collaborative manipulator. While picking worked well in the lab setting, it turned out to be challenging to transfer this skill to the mobile STAMINA robot with a larger manipulator in the evaluation setup. Hidden assumptions had possibly been made in the lab setting, which might have been violated in the evaluation setup. Furthermore, different environmental conditions, such as direct sunlight, and different calibration procedures were also limiting factors for performance. However, in hindsight, it is clear that these shortcomings could only have been discovered during experiments in the real production environment. Another complicating factor was that time for physical presence in the production environment was limited and remote setup support turned out to be challenging. As a result, three valuable lessons were learned from these experiences: 1) to work with the real evaluation platform as early as possible, 2) to involve evaluation engineers in the development of the testing environment, and 3) to test methods in a variety of setups to ensure their robustness to changes, and to limit the chance of hidden assumptions.

6 Discussion and Conclusions

In this paper and in previous work [18], the STAMINA project presented the results of using a mobile autonomous robot for a kitting task at the car manufacturer PSA Peugeot Citroën. Throughout this work, the project's goals were to

1. push the technology and scientific expertise further in the direction of the use-case,

2. identify problems and challenges that could not have been identified without actually approaching and attempting to solve the use-case,
3. quantify what works and how well it works,
4. identify what does not work and attempt to explain why,
5. outline next steps to overcome the identified problems, and
6. provide a baseline for performance comparison with future systems.

The hardware and software for the robot was already conceptualized in 2014, based on experiences from earlier projects. However, a considerable amount of development happened during later test sprints in experimental areas at PSA where the approach was validated against real needs. This led to numerous new insights and scientific challenges. The experimental results obtained from using the STAMINA robot system in the use-case of the final test sprint were presented here, and the scientific insights and advances were summarized in multiple papers.

While more robots would be needed in order to reach PSA's official cycle time of (approximately) 60 sec, the experiments nevertheless demonstrate the successful operation of such a robot in its proper environment: the robot was completely integrated with the manufacturing execution system (MES), it was able to plan and execute its tasks, the logistic planner held all relevant information for task planning and execution, and the system was able to manage and recover from various unexpected errors. Overall, the system was able to fetch the right parts at the right time from the warehouse and deliver them to the necessary exit point. It is easy to imagine a mobile robot with a smaller mobile base fetching and handing tools and parts to the line workers. However, running the robot in a real environment also made its shortcomings apparent, and identified problems that were not evident from the start: the variety of different parts in the warehouse is very large, and all known robot grippers are limited to only a very small subset of objects and object properties (e.g., fragile, shiny, transparent, unbalanced, heavy, etc.). In fact, it is tempting to conclude that one of the present weaknesses in such a robotic kitting system is the lack of a sufficiently versatile gripper. In practice, one potential solution would be to use a set of grippers with a tool changer.

Another challenge that is also related to the grippers is how the parts are packed (see Fig. 16). Typical packaging material includes plastic wraps, cardboard separators, bulk storage, stacking, etc. Packaging material needs to be removed, some objects come in plastic bags, some are packed tightly in Styrofoam, etc. Finally, task constraints (e.g., limitations on allowable grasps, likely part entanglements, placing constants, etc.) can further complicate the picking, as depicted in Fig. 17. One cornerstone of this project was the safety of the system on the shop floor. As

Fig. 16: Picking challenges due to part packaging (from top left to bottom right): empty boxes need to be removed; cardboard dividers need to be removed; white parts in see-through plastic bags; plastic covers; another example of cardboard separators; rightly stacked heat shields; decorative parts packed in Styrofoam; wheel caps in plastic bags.

the system was supposed to work in the vicinity of humans without fences being required, current regulations require strict limits on the speed of the mobile platform and the manipulator. An initial idea was to use the laser-range scanners of the mobile platform to identify if a human was in the direct vicinity and then to adapt the robot arm speed accordingly. However, it turned out that the stability of some of the grasps of the robot gripper was critical due to hardware limits: through the experiments it became apparent that the 3 finger Robotiq gripper was too small for some of the large parts, and many of the parts from the kit in Fig. 12 could barely be picked. Furthermore, most of these parts were also very heavy, and the high forces under fast arm movements can easily break the fingers of the robot's gripper. In that sense, as already outlined above, choosing a suitable gripper is presently a great challenge. Since it was not possible to ensure the absence of humans in the potential ballistic direction, the arm speed had to be limited. However, limiting the robot's speed also resulted in a large cycle time meaning that a large number of robots would be required in order to reach the PSA's cycle time of approximately 60 sec per kitting box. Possibilities worth exploring are to make the arm movements dependent on the object weight or to possibly use a force-torque sensor in the wrist to control the maximum forces in the gripper to prevent uncontrolled loss of the object in the gripper. For future installations at PSA, safety requirements need to be further explored based on new upcoming regulations and new and upcoming robotic hardware.

In this work, all robot control was based on the skills and a planner that automatically selected these skills based on a world model. After the tests summarized

Fig. 17: Example of the various picking criteria that influence the choice of hardware (robot, gripper, vision) and grasping technique (left): the alternator should not be picked by its rotor axis (right), as shown in the SOP of the kitting technician.

above, a STAMINA system with similar capabilities was also demonstrated in various other environments (e.g., in a different factory environment). The setup of the system was always the same: once the logistic world model was "programmed", the system worked directly, and no changes to the software of the STAMINA system were required. Furthermore, SkiROS-based high-level robot control was tested on multiple mobile manipulators with different hardware. While hardware variability was handled on the skill primitive level through ROS, high-level robot control, task planning, and vertical embedding of the robot into the MES worked directly out of the box.

The recent 2018 ARIAC competition [25] organized by NIST reflects the present interest in high-level planning for automated kitting. Members of the STAMINA team participated in the competition with the STAMINA approach ranking third [25]. In terms of effort, it took three days to pass the qualification threshold using the same approach described here. During the competition it became apparent that the STAMINA scenario differed from the one in the ARIAC 2018 competition: in STAMINA, kitting orders were never called back or changed, and never put on hold in favor of higher-priority kitting orders. In the ARIAC competition, these were normal situations. To handle these situations, the STAMINA approach was extended to include predefined short-term behaviors which were modeled with behavioral trees [33]. Including these extensions, the entire competition required one week of work.

Finally, the skills implemented on the robot reflected how *skilled* the robot was in picking, placing and driving. For example, to improve the picking of the engine pipe, only improvements to the pick skill were required while the rest of the system was not affected. Furthermore, adding a skill and skill primitive (e.g., `gripper_oc`) is straight forward [32], and the planner is able to immediately make use of such additional capabilities [33].

In ongoing work within the EU project SCALABLE 4.0, the STAMINA robot is currently being extended for use as a mobile and *skilled* manipulator resource for highly modular manufacturing lines. The new robot tasks include packing and assembly tasks, and the robot is meant to execute these tasks wherever needed on the manufacturing line.

Acknowledgements

This work was partially funded by the European Union's 7th Framework Programme under grant no. 610917 (STAMINA) and by the European Union's Horizon 2020 research and innovation programme under grant agreement 723658 (ScalABLE 4.0).

References

[1] H2020 project ScalABLE4.0: Scalable automation for flexible production systems. https://www.scalable40.eu. Grant agreement no 723658.

[2] R. Arrais, M. Oliveira, C. Toscano, and G. Veiga. A mobile robot based sensing approach for assessing spatial inconsistencies of a logistic system. *Journal of Manufacturing Systems*, 43:129–138, 2017.

[3] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language reference, 10 February 2004. Available at `http://www.w3.org/TR/owl-ref/`.

[4] Valeria Borodin, Alexandre Dolgui, Faicel Hnaien, and Nacima Labadie. Component replenishment planning for a single-level assembly system under random lead times: A chance constrained programming approach. *International Journal of Production Economics*, 181:79 – 86, 2016. SI: ISIR 2014.

[5] Nils Boysen, Simon Emde, Michael Hoeck, and Markus Kauderer. Part logistics in the automotive industry: Decision problems, literature review and research agenda. *European Journal of Operational Research*, 242(1):107 – 120, 2015.

[6] M. Crosby, M. Rovatsos, and R. Petrick. Automated agent decomposition for classical planning. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 46–54, 2013.

[7] M. Crosby, F. Rovida, V. Krueger, and R. Petrick. Integrating mission and task planning in an industrial robotics framework. In *Proceedings of the International Conference on Automated Planning and Schedul ing (ICAPS 2017)*, Pittsburgh, PA, USA, June 2017.

[8] M. Crosby, F. Rovida, M.R. Pedersen, R. Petrick, and V. Krueger. Planning for robots with skills. In *Proceedings of the ICAPS 2016 Workshop on Planning and Robotics (PlanRob)*, 2016.

[9] B. Grossmann and V. Krueger. Continuous hand-eye calibration using 3d points. In *Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN)*, 2017.

[10] B. Grossmann and V. Krueger. Fast view-based pose estimation of industrial objects in point clouds using a particle filter with an ICP-based motion model. In *Proceedings of the 15th IEEE International Conference on Industrial Informatics (INDIN)*, 2017.

[11] B. Grossmann, M. Siam, and V. Krueger. Comparative evaluation of 3d pose estimation of industrial objects in rgb pointclouds. In *Proceedings of the International Conference on Vision Systems (ICVS)*, 2015.

[12] D. Holz, A.Topalidou-Kyniazopoulou, F. Rovida, F.R. Pedersen, V. Krueger, and S. Behnke. A skill-based system for object perception and manipulation for automating kitting tasks. In *Proceedings of the 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Luxembourg, 2015.

[13] D. Holz, A. Topalidou-Kyniazopoulou, F. Rovida, M.R. Pedersen, V. Krueger, and S. Behnke. A skill-based system for object perception and manipulation for automating kitting tasks. In *Proceedings of the 20th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2015.

[14] D. Holz, A. Topalidou-Kyniazopoulou, J. Stueckler, and S. Behnke. Realtime object detection, localization and verification for fast robotic depalletizing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[15] International Organization for Standardization (ISO), www.iso.ch. *ISO/TR 16982:2002 Ergonomics of human-system interactionUsability methods supporting human-centered design*, 2002.

[16] International Organization for Standardization (ISO), www.iso.ch. *ISO 9241-210:2010 Ergonomics of Human System Interaction - Part 210: Human- centred design for interactive systems*, 2010.

[17] Niki Kousi, Spyridon Koukas, George Michalos, Sotiris Makris, and George Chryssolouris. Service oriented architecture for dynamic scheduling of mobile robots for material supply. *Procedia CIRP*, 55:18 – 22, 2016. 5th CIRP Global Web Conference - Research and Innovation for Future Production (CIRPe 2016).

[18] V. Krüger, A. Chazoule, M. Crosby, A. Lasnier, M.R. Pedersen, F. Rovida, L. Nalpantidis, R. Petrick, C. Toscano, and G. Veiga. A vertical and cyber-physical integration of cognitive robots in manufacturing. *Institute of Electrical and Electronics Engineers. Proceedings*, 104(5):1114–1127, 2016.

[19] Eugene Levner, Leonid Meyzin, and Frank Werner. Hierarchical scheduling of mobile robots in production-transportation supply chains. *IFAC Proceedings Volumes*, 42(4):786 – 791, 2009. 13th IFAC Symposium on Information Control Problems in Manufacturing.

[20] J. Luft and H. Ingham. The johari window, a graphic model of interpersonal awareness. In *Proceedings of the western training laboratory in group development.*, 1955.

[21] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins. PDDL – The Planning Domain Definition Language (Version 1.2). Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, October 1998.

[22] George Michalos, Niki Kousi, Sotiris Makris, and George Chryssolouris. Performance assessment of production systems with mobile robots. *Procedia CIRP*, 41:195 – 200, 2016. Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future - Proceedings of the 48th CIRP Conference on Manufacturing Systems.

[23] J. Nielsen. *Usability Engineering*. Nielsen Norman Group, 1993.

[24] T. Niemueller, E. Karpas, T. Vaquero, and E. Timmons. Planning competition for logistics robots in simulation. In *Proceedings of the ICAPS 2016 Workshop on Planning and Robotics*, 2016.

[25] NIST. Agile robotics for industrial automation competition (ariac), 2018. `https://www.challenge.gov/challenge/ariac/` `https://www.openrobotics.org/blog/2018/5/31/ariac-2018-finals-results-announced`.

[26] M.R. Pedersen and V. Krueger. Automated planning of industrial logistics on a skill-equipped robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop Task Planning for Intelligent Robots in Service and Manufacturing*, 2015.

[27] M.R. Pedersen, L. Nalpantidis, R.S. Andersen, C. Schou, S. Bœgh, V. Krueger, and O. Madsen. Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing*, 37:282–291, 2016.

[28] A. S. Polydoros, B. Grossmann, F. Rovida, L. Nalpantidis, and V. Krueger. Accurate and versatile automation of industrial kitting operations with skiros. In *17th Annual Conference Towards Autonomous Robotic Systems (TAROS)*, pages 255–268. Springer, 2016.

[29] A. S. Polydoros, L. Nalpantidis, and V. Krueger. Real-time deep learning of robotic manipulator inverse dynamics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[30] J. Roewekaemper, B.Suger, W. Burgard, and G. D. Tipaldi. Accurate localization with respect to moving objects via multiple-body registration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3741–3746, 2015.

[31] J. Roewekaemper, M. Ruhnke, B. Steder, W. Burgard, and G. D. Tipaldi. Automatic extrinsic calibration of multiple laser range sensors with little overlap. In *Proceedings of the International Conference on Robots and Automation (ICRA)*, pages 2072–2077, 2015.

[32] F. Rovida, M. Crosby, D. Holz, A. Polydoros, B. Großmann, R.P.A. Petrick, and V. Krueger. *SkiROS—A Skill-Based Robot Control Platform on Top of ROS*, pages 121–160. Springer International Publishing, Cham, 2017.

[33] F. Rovida, B. Grossmann, and V. Krueger. Extended behavior trees for quick definition of flexible robotic tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[34] F. Rovida, V. Krueger, C. Toscano, G. Veiga, M. Crosby, and R. Petrick. Planning for sustainable and reliable robotic part handling in manufacturing automation. In *Proceedings of the ICAPS 2016 System Demonstration Session*, 2016.

[35] F. Rovida and V. Krüger. Design and development of a software architecture for autonomous mobile manipulators in industrial environments. In *IEEE International Conference on Industrial Technology (ICIT)*, 2015.

[36] The Partnership for Robotics in Europe SPARC˙ Robotics 2020 multi-annual roadmap. Technical report, EU-Robotics, 2015.

[37] César Toscano, Rafael Arrais, and Germano Veiga. Enhancement of industrial logistic systems with semantic 3d representations for mobile manipulators. In Anibal Ollero, Alberto Sanfeliu, Luis Montano, Nuno Lau, and Carlos Cardeira, editors, *ROBOT 2017: Third Iberian Robotics Conference*, pages 617–628, Cham, 2018. Springer International Publishing.