D2.1: Internal Requirements Document

30th June 2022

Project funded by the European Commission within the Horizon 2020 Programme		
Dissemination Level		
PU	Public	**X**
CO	Confidential, only for members of the consortium (including the Commission Services)	□
CL	Classified, as referred to in Commission decision 2001/844/EC	□

Deliverable number:	**D2.1**
Deliverable name:	Internal Requirements Document
Work package:	WP2: Requirements
Lead WP:	Raj Kamal (Meritor)
Lead Task:	Raj Kamal (Meritor) and all WP2 Task Leaders

Contents

Document Revision History

Date	Issue	Author/Editor/Contributor	Summary of main change
23/04/2022	V1	Raj Kamal and Andrea Micheli	Initial draft comprising the use-case documents devised during WP2 by all involved partners
31/05/2022	V2	All WP2 Partners	Revised use-cases presentation and completed demonstrator sections
28/06/2022	V3	All WP2 Partners	Final proofreading
31/01/2023	V4	Andrea Micheli	Added requirements collected for each use-case
30/03/2023	V5	Andrea Micheli	Final proofreading after discussion at physical meeting

Abstract

Automated Planning and Scheduling is a central research area in AI that has been studied since the inception of the field and where European research has been making strong contributions over decades. Planning is a decisionmaking technology that consists in reasoning on a predictive model of a system being controlled and deciding how and when to act in order to achieve a desired objective. It is a relevant technology for many application areas that need quick, automated and optimal decisions, like agile manufacturing, agrifood or logistics. Although there is a wealth of techniques that are mature in terms of science and systems, several obstacles hinder their adoption, thus preventing them from making the footprint on European industry that they should make. For example, it is hard for practitioners to find the right techniques for a given planning problem, there are no shared standards to use them, and there is no easy access to expertise on how to encode domain knowledge into a planner.

The AIPlan4EU project will bring AI planning as a first-class citizen in the European AI On-Demand (AI4EU) Platform by developing a uniform, user-centered framework to access the existing planning technology and by devising concrete guidelines for innovators and practitioners on how to use this technology. To do so, we will consider use-cases from diverse application areas that will drive the design and the development of the framework, and include several available planning systems as engines that can be selected to solve practical problems. We will develop a general and planner-agnostic API that will both be served by the AI4EU platform and be available as a resource to be integrated into the users' systems. The framework will be validated on use-cases both from within the consortium and recruited by means of cascade funding; moreover, standard interfaces between the framework and common industrial technologies will be developed and made available.

Executive summary

In this deliverable, we will report all the use-cases provided by partners of the AIPlan4EU consortium. For each use-case, the general context is presented, followed by an in-depth explanation of the planning application. Moreover, each use-case highlights its technical and business KPIs and assesses the expected impact of the introduction of planning technology both from a business and a technical point of view.

From the use-cases descriptions, we derive a set of high-level requirements for the Unified Planning Framework and we describe the expected demonstrations that will be output of the project. These requirements and definitions are the input for work packages 3,4,5 and 6.

1. Introduction

One of the overarching goals of the AIPlan4EU Project is to make planning technology usable for companies, SMEs and Innovators. For this reason, the project was designed to follow an agile approach grounded in several use-cases. The project counts on 7 use-case provider partners, and devotes a substantial amount of the cascade funding resources to elicit new and interesting use-cases.

This deliverable focuses on the use-cases elicited and developed by partners of the consortium, while D2.2 [M30] will be an extended and updated version of this document containing also the use-cases elicited by means of open-calls. In particular, the use-cases and the partners covered in this deliverable are reported in the table below.

Task ID	Use-Case Provider Partner	Research Partner	Use-Case Title
T2.1	Trasys (TRA)	FBK	Space Domain: Generation and Consolidation of Activity Plans
T2.2	Agrotech valley Forum (ATV)	DFKI	Agriculture Domain: Silage Maize Harvesting
T2.3	Meritor (MER)	ORU	Flexible Manufacturing
T2.4	Magazino (MAG)	ORU	Logistics Automation: Offline Design Aid Tool and Runtime Reactive Planner for Behavior Trees
T2.5	Easy Mile (EM)	CNRS	Shuttle Fleet Management: Mission Order Assignment
T2.6	Procter & Gamble (PGBS)	UNIROMA1	Automated Experiment Design: Automation of Consumer Goods Testing
T2.7	Saipem (SAI)	FBK	Subsea Robotics: HyDrone Planner and Replanner

The work reported in this deliverable has been performed in the context of WP2 - Requirements and it encompasses outputs generated by all the tasks. The work has been organized in subsequent steps throughout the first year of the project. The steps we followed are reported below.

1. **Use-case presentation**: Each partner presented their use-case general idea and context to all the other partners in WP2. The Technical partners gave feedback and asked questions to identify the characteristics of the various planning problems that arose.
2. **Development of use-case templates**: In order to uniform the work among the many tasks composing WP2, the team, lead by Meritor and FBK, developed a general template to be followed by all the use-cases developed within the project. The same template has been used in WP10 as the basis for the Open Calls for use-Cases.
3. **Iterative refinement of use-cases**: Each use-case provider partner was paired to a research partner according the table above(we fully respected the task allocation reported in the Grant Agreement) and together they developed the use-case in the full details and defined the KPIs, the expected results and narrowed down the planning problem characteristics. The result of this work is reported in the following sections (one for each use-case).

Structure of the document

This document is structured as follows. The next section presents the template that has been developed in step 2, while Section 3 reports, for each use-case, the full description of the use-case. Section 4 outlines the directions being followed for the realization of the demonstrators of each use case and Section 5 draws the conclusions.

2. Use-case templates

Each use-case will be described according to the standard template outlined below.
Each section (in bold) is associated with a set of questions to be answered by the use case provider and the template was designed to provide enough information to a planning expert to identify a suitable planning-based solution; moreover, we asked each use-case provider (and each open-call applicant) to identify impacts and KPIs for the use-case and identify relevant data and integration constraints.

1 **Context**: describes the context of the use-case, in particular:
- Which is the application domain?
- How are the operations handled currently? Is there an established workflow?
- What are the drawbacks of the current solution (if any)?

2 **Planning Application**: planning is a technology that helps you in automating or optimizing some decisions given a predictive model of a system / situation. Here we clearly identify the key decisions that should be automated/optimized in the use-case.
- What are the key problems that you plan to address with planning technology?
  - Highlight which kind of decisions shall be automated or optimized
- How do you measure the quality of such decision/automation? Why is the status quo not satisfactory?
- Which kind of data is relevant for taking the decisions to be automated /optimized?
  - Highlight what is “constant” (i.e., it does not change between successive decisions) and what is “contingent” (i.e., non-predictable data that is important to take an informed decision)
- How often is planning invoked? Which actor/software/rule invokes the planning?
  - E.g., planning is done once a day to decide the shift activities
  - E.g., planning is triggered when a discrepancy is observed w.r.t. the planned course

3 **Introductory Example**: we provide a detailed example case listing the needed data and the expected decision outcome.

4 **Impact**: we describe the expected impacts of using planning technology in the use-case from different perspectives.

4.1 <u>Business Impacts</u>
- What are the business impacts of automating/optimizing the decisions?
- How does the quality measure defined in the previous section map to the business? E.g.:
  - by automating the picking from shelves, we can reduce the operational costs by …
  - by optimizing the management of agricultural practices, we can reduce soil compaction that is bad because…)
- How do you define a satisfactory measurement from the business perspective? (e.g., we aim at reducing soil compaction by 10% because…)

4.2 <u>Other Impacts</u>
- Is there any other impact besides business? E.g. environmental, societal…

5 **Measures of Success**: we describe how, concretely, one can measure the success of a planning integration in this use-case.
- KPIs: what are key indicators and their thresholds?
- Expected performance: computational speed, solution quality…
- How can one make these measures?

6 **Planning Integration**: we describe how planning will fit in the workflow/pipeline and which consequences/requirements this integration poses. Moreover, if integration with any technology is relevant, please provide the details.
E.g., if we are planning workforce activities, we need ways to avoid alienation and to properly communicate with operators.
E.g., If the flow of goods in a warehouse is governed by a certain WMS system, integrating the planner with such technology is important and requirements for such an integration must be considered.

7 **Example and Evaluation Data**: we report (or provide pointers to external files to be delivered) the data needed to experiment with the use-case. We provide some small-scale examples as well as real-sized problems. We describe how to interpret the data (e.g., you can find the operation timings in table XXX and the set of locations is reported in table YYY).

8 **Requirements:** A formalization in tabular form of the requirements collected from the use-case. The table format adopted for all the use-cases is depicted below.

<table>
<tr><th colspan="2">[ID / Title of the requirement]</th><th>Req. Type</th><th>Verif Approach</th></tr>
<tr><td rowspan="2"></td><td colspan="3">[Description]</td></tr>
<tr><td colspan="3">[Comment]</td></tr>
</table>

3. Internal Use-Cases

T2.1 Space-domain

The ‘Planning for Space’ use case targets the automation of the planning process in the context of multi-asset human-robotic missions as prepared by the National Space Agencies, the European Commission and the European Space Agency. Typical examples are the ExoMars mission for Mars exploration, the Mars Sample Return mission as well as the HERACLES mission for moon exploration and exploitation.

In this area there is currently no generic and reusable European reference architecture. The current solutions are proprietary and mission specific. It is therefore necessary to extend various tools and components of the existing Mission Operations Infrastructure, such as the 3DROCS TRASYS tool, with automated operations planning features. In particular 3DROCS supports robotics operations in Telemanipulation, Interactive Autonomy and Autonomy modes and it is at the basis of the rover planning tools in the ExoMars 2020 operations centre.

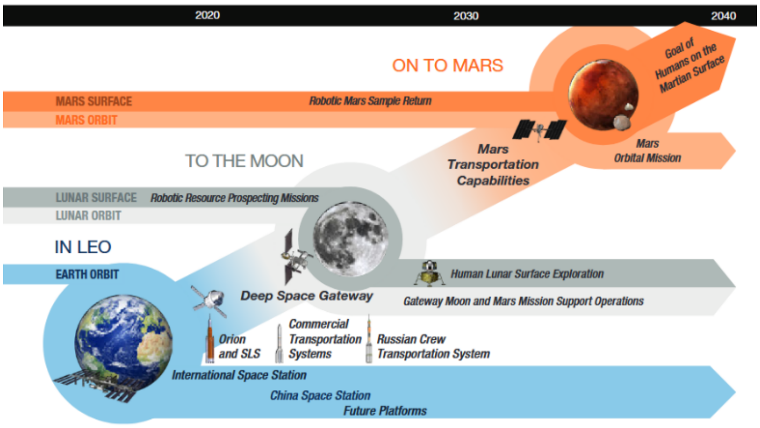
At the basis of the Activity Planning process resides a) the decomposition of a robotic system in a set of subsystems and the modelling of their behavior and b) the concept of the Activity and the Activity Library provided by the manufacturer of the robotic system: rover and instruments operations are organized in terms of Activities (Actions and Tasks) associated to one or several subsystems. An Activity drives the corresponding subsystem from an initial state to a final one with associated resources consumptions (duration, power and memory mass) often dependent on the location and the time the Activity is executed on the planetary surface. The objective of the Activity Planning process is to produce a validated Consolidated Activity Plan to be uploaded to the robotic system for execution. The Consolidated Activity Plan is an aggregation of Partial Activity Plans proposed by the engineers and the scientists participating in the mission.

In conclusion, in the context of the AIPlan4EU: Planning for Space Use Case, the objective of TRASYS is to extend the 3DROCS functionality to interface with the UPF to access and evaluate state-of-the art planners in support to multi-asset human-robotics space operations. The evaluation will be based on data already generated and expected to be generated in the context of the ExoMars mission preparation. The evaluation criteria include the decrease of the duration of the operations preparation Planning cycle and the efficiency of the resulting plans in terms of science return and resources consumption.

Context

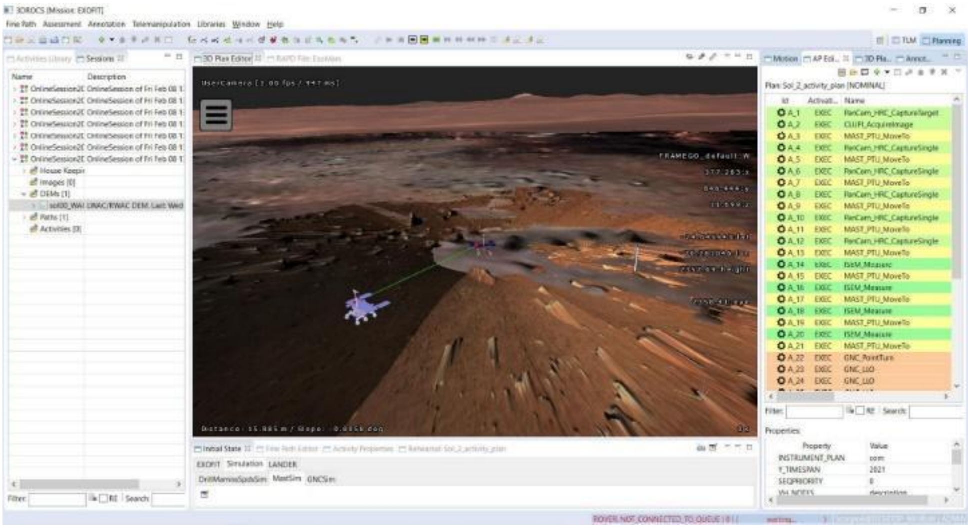
Application domain

The Global Exploration Roadmap (GER) foresees the stepwise advancement of humankind into our solar system within the next 20 years with advancement towards the Moon within the next ten years to mature knowledge and technologies for human missions to Mars in the next twenty years. In this global vision, robotic missions precede human explorers to the Moon, near-Earth asteroids, and Mars. Humans advance beyond Earth orbit and ISS to a lunar orbital station and eventually human lunar surface missions. In parallel, Mars is continuously explored robotically with humans on the Martian surface towards the 2040ies.



Currently, the ESA Mission Operations Infrastructure (MOI), at ESOC, encompasses a number of data systems that have been developed gradually in the past decades with focus on providing the capabilities, required for the operations of satellites in Earth Observation and Science Programs. This has led to the formation of a generic multi-mission reference ground segment architecture, based on international standards and an infrastructure for mission planning, simulation, monitoring and control, data dissemination and other capabilities related to satellite mission operations.

The analysis of the operational concept of the new human-robotics mission types involves elements of distributed, multi-asset, multi-centre operations where individual cooperating assets of larger systems will be operated by different entities (e.g., the owner of the asset) adding considerable complexity to the mission operations concepts and equally on the operations planning and the supporting mission data systems. In this area there is currently no generic and reusable European reference architecture. The current solutions are proprietary and mission specific. It is therefore necessary to extend various components of the existing MOI, including automated operations planning, with corresponding features, in order to be able to support these new multi-asset human-robotic mission types based on the works performed so far by the ESA and in particular on 3DROCS[1].



3DROCS, developed by TSYS, supports robotics operations in Telemanipulation, Interactive Autonomy and Autonomy modes and it is at the basis of the rover planning tools in the ExoMars 2020 operations centre.

[1] Luc Joudrier, Konstantinos Kapellos and Kjetil Wormnes: 3d Based Rover Operations Control System. http://robotics.estec.esa.int/ASTRA/Astra2013/Papers/joudrier_2824438.pdf

Current Workflow

The typical Activity Planning workflow (also referred as tactical planning) involves a well-defined set of steps presented here below; at each step we detail only the elements that are relevant to the AIPlan4EU activity. There are:

- **Telemetry Acquisition and Processing**: the telemetry packets are received and processed to the appropriate level to generate products to be further analysed by the engineering and science teams.
- **Engineering and Science Data Assessment**: the rover engineers and the scientists, in parallel, analyse the rover system status, the followed path, the payload status, the imagery and the instruments products. At this point the utilization of the downloaded *Execution Report* allows the teams to assess which Activities have been really executed and the amount of the consumed resources. In addition, the analysis of the downloaded data allows to establish the *Initial State* to be considered as the starting point for the planning operations. The Execution Report and Initial State entities (definition, model and format) are of particular interest in our use case.
- **Engineering and Science Planning**: the engineers and the scientists, based on the assessment results, in parallel, define the Partial Engineering Activity Plans and the Partial Science Activity Plans they would like the rover executes in the next period. Partial Engineering and Science Activity Plans are manually created by composing Activities using dedicated MMIs. The prepared Partial Plans are submitted to a central Activity Planning tool for further consolidation. The concept of the *Partial Activity Plan* is central in our use case.
- **Activity Plan Consolidation**: during this step an Activity Planner integrates all the submitted Partial Science and Engineering Activity Plans and schedules them in a semi-automatic way in a *Consolidated Activity Plan* so that constraints and resources are respected.
- **Activity Plan Validation and Uplink**: this step covers the validation of the Consolidated Activity Plan by an operational simulator and its uplink. The validation process creates a simulated Execution Report and a simulated Final State with which the real data will be compared in the next Engineering and Science Data Assessment phase.

Note that this workflow is bounded by the time in between the reception of the Rover telemetry and the dispatching of the Activity Plan in the following contact window (about 12 hours) and each step shall be completed within strict deadlines.

Drawbacks of the Current Solution

The main drawbacks in the current solution are the following:

- **Manual preparation of Partial Engineering and the Science Activity Plans**. To this end, engineers and scientists are obliged to be aware of several engineering low level constraints (e.g. set the robotic system to a state compliant to their objectives, avoid the use of subsystems that create conflicts during operations, be aware about the resource consumption of their objectives). As a result, the Partial Planning step requires a significant amount of time to be completed and very often provides invalid Partial Plans that are later rejected.
- The current solution uses an adapted version of the FlexPlan satellite operations proprietary planning tool to support the consolidation of the Partial Activity Plans to a final valid Consolidated Activity Plan to be uploaded for execution. The drawbacks of this solution are, at one hand, the proprietary character of the tool that **does not provide the required flexibility** to support mission preparation activities such as field testing etc. and, on the other hand, the nature of the tool that **targets satellite model of control** based on TC sequences than on Activities and Activity Plans.

A complementary drawback in the current approach is the **non-collaborative** aspect for the elaboration of the Activity Plans. As explained in section 2.2, the process is sequential and individually performed resulting on many rejected Partial Activity Plans at the consolidation phase while a more agile approach (continuous submission and consolidation of Activity Plans) could provide better results.

Planning Application

Key problems

If we adopt the **Operator's point of view,** we will use planning technology to address the following problems:

- **Elaborate Partial Activity Plans automatically** allowing the operator to express his objectives as high-level goals and this, if possible, following a mixed initiative approach where the operator is informed and participates in the decisions of the planner.
- **Consolidate Partial Activity Plans automatically** using open source and adapted to the domain technology. Here also a mixed initiative planning approach is a nice to have objective.
- **Integrate the planning technology into an operational tool:** the objective is to interface the AIPlan4EU framework with 3DROCS in order to provide the operators access to the available planning technology.

If we adopt the **Robotic System Point of view,** we plan to interface the planning technology with our simulator (see section 7.9) in order to:

- **Elaborate Activity Plans on-board:** currently the simulator executes predefined Activity Plans with limited planning and replanning (repair) capabilities.
- **Execute the elaborate Activity Plans:** currently the simulator interprets and executes plans described in 'exomars' like description language and in a reactive event-based environment. These capabilities can be further developed guaranteeing their genericity.

Quality measure

The quality level of Activity Plans is characterized by the amount of the expected science data generation and the efficient management of the robotic system critical resources, i.e. the time, the power and the memory mass consumption. For a given activity plan, the ROCS (Rover Operations Control System) allows to quantify these measures: number of generated science products downloaded, operations time in a given operations period (e.g. a sol), and finally the state of charge of the battery and the memory mass occupancy at the end of the operations period. We expect that the use of automatic planning technologies will bring a significant improvement of these measures (of at least 10%) allowing the comparison of different planning solutions and measuring the use-case satisfaction.

Relevant Data for decision making

The starting and invariant point of our use case is the Functional Analysis of the robotic system that results on:

- The decomposition of the global system into subsystems and the description of their behavior in a formal model. The model represents the states of each subsystem as well as the consumed resources per state (see section 7.3),
- The identification of the Activities per subsystem characterized either as Actions (atomic Activities) or Tasks (logical and temporal composition of Actions). The Activities change the states of the subsystem(s) to which they belong.
- The specification of constraints between subsystems and between Activities (for example it is not possible to drill while traveling),
- The identification of high-level goals function of a given mission.

During the Planning process the following data may change:

- The state of the system before each planning cycle.
- The time and the location for which the Activity Plan is produced as the environmental conditions (temperatures, fluxes, etc) highly depend on these data.
- Finally, the topological and morphological aspects of a continuously discovered environment is important with the additional difficulty that it is generally partially known or even unknown.

Business case

If we adopt the **Operators Point of View**, i.e. the impact on the on-ground planning process, the business impact includes the following considerations:

- Planning time is precious as communication windows impose short planning periods: it is expected that the automatic planner reduces the planning time.
- Automatic consolidation of plans (including mixed initiative) allows for collaboration from the user home basis.
- The automatically generated Activity Plans are expected to optimize the resources consumption (time and power) and therefore to maximize the science and the engineering feedback.

If we adopt the **Robotic System Point of View** the business impact:

- Allows to operate on remote planets without frequent communications,
- Maximizes the duration of operations per day and therefore the operations science returns.

The **business impact can be evaluated** considering**:**

- The improvement of planning time with 'one planner': provide an estimate
- The Improvement of planning time with 'collaborative planning': provide an estimate
- The Improvement of resources exploitation: user defined plans are conservative: provide an estimate
- The Improvement of resources exploitation on-board when replanning: provide an estimate

Planning Integration

The integration of the AIPlan4EU planning technologies into 3DROCS includes:

- The update of the Activity and Activity Plans models to capture the AIPlan4EU planning technology needs (Activities priorities, min/max start time, min/max duration, …).
- The enhancement of the Planning MMIs to specify the newly introduced elements,
- The development of the necessary API to interface the UPF to submit planning problems and receive and visualize the planning solutions.

Example and Evaluation Data

Data and Tools to experiment the use-case

The **data needed to experiment with the use case** are the following:

- The Activity Template Library (ATL) as result of the Functional Analysis of the system (see section 7.3).
- The Initial State of the system at the moment each planning problem is applied (see section 7.4),
- The Activity Plan (including goals), specified as a composition of Activities (goals) (see section 7.5),
- The Final State of the system at completion of the simulation of the automatically generated plan (see section 7.6),
- The Activity Plan Execution Report reporting the evolution of the execution of the Activities as well as the consumed resources at Activity and at Activity Plan level (see section 7.7).

The **tools needed to experiment with the use case** are the following

- The 3DROCS for the ATL, State Vectors and Activity Plans Management as well as for their visualization (see section 7.8),
- The 3DROV simulator to simulate an Activity Plan and provide the Activity Plan Execution Report (see section 7.9).

Data and Tools to evaluate the use-case

The **data needed to evaluate the use case** are the following:
- The Activity Plans as manually specified by the operators fulfilling the same objectives,
- The corresponding Final States and Execution Reports that indicate the evolution of the execution of the manually prepared Activity Plan and the consumed resources.

The **tools needed to evaluate the use case** are the following:
- The 3DROCS for the visualization of all data involved in the comparison as for example, the visual comparison of the Execution Reports.

Activity Template Library

The Activity Template Library (ATL) is provided in the form of an XML file with the following structure:
- All information is included into the <ATLmodel> and </ATLmodel> tags,
- **Mission Data Types** are encapsulated into the <missionDataTypes > and </missionDataTypes > tags,
- **Subsystem information** is encapsulated into the <subsystem > and </subsystem > tags,
- The **groups of subsystems** information are encapsulated into the <subsystemGroups > and </subsystemGroups > tags,
- The **Template Activities** are included into the <templateActivities> and </templateActivities> tags.

Moreover, each ATL is provided with a specific version to ease the update and validation processes. The version of the ATL is in the **<version/>** element.

3DROCS Control Station

3DROCS is a Robotic Control Station developed by TRASYS that allows the operating of a robotic system in Telemanipulation, Interactive Autonomy and Autonomy modes. In the AIPlan4EU activity, 3DROCS is the target Control Station in which the planning technologies will be integrated. In particular 3DROCS provides the functionality to support Planning & Validation, Monitoring & Control and Assessment. We further develop here below the Planning aspects in Interactive Autonomy and Autonomy operations modes. Let us focus on the planning capabilities in Autonomy operations modes that consist in the specification and the validation of Activity Plans and of Goal Plans.
- **Activity Plans** are specified as a logical and temporal composition of robotic activities. They may be the result either a) of a user specification that selects Activities and synthesizes them with control primitives (e.g. sequence, parallelism, loops, variables evaluation) or b) of an on-ground planning system that decomposes a user defined Goal to an Activity Plan. These plans, before requesting their execution, shall be checked at least with respect to the validity of their sequencing, the availability of the required system resources such as time, power and memory mass and with respect to their ability to satisfy the operators objectives.
- **Goal Plans** are specified as a logical and temporal composition of Goals. Goal plans can be validated by the on-ground Activity Planner to guarantee that if decomposed on-board will lead to valid Activity Plans.

Therefore, the Robotics Operations Station infrastructure shall have a model of the entities required by the operator to Plan and Validate robotic operations. The Robotics Operations Station tools shall be conceived as MMIs reflecting the instantiation of the models for the current mission.
The following models shall be considered:
- **Models of the robotic assets**. It includes:
    - The 3D model of the robotic system including its Kinematics and possibly its dynamics,

    - The 'State' of the robotic asset that fully defines the state of the robotic system (e.g. the platform location, the mast and the joint values, that status ON/OFF of a subsystem are typical elements of a robotic system state vector.
- **Models of the environment**: it consists in the 3D morphological representation of the environment including as well as the necessary planet parameterisation that allows faultless mapping, projection and distance measurements.
- **Models of the Activities**: the set of the on-board Activities that can be used from the ground control together with their parameters as well as their influence on the robotic system state and resources consumption. They are organized in an Activity Template Library (ATL) generally provided by the robotic asset constructor.
- **Models of the 3D Annotations**: we consider here the following entities:
- The Targets: that indicate target positions to be reached by a rover, Paths are defined as an ordered sequence of waypoints: they indicate target positions to be reached by a rover together with a set of intermediate positions to be visited. This allows it to ensure that when the rover is traveling it will pass from areas considered as safe. Labels: allow to name locations in the environment to facilitate the discussion between operators. Frames: that specify reference locations in the scene (e.g. the landing frame), Areas: that define areas of interest or forbidden area.
- A set of tools shall be considered to allow the operator to plan and validate robotic operations:
    - 3D Rover Visualization and Planning (3D-RVP): that includes the models of the robotic assets and the environment,
    - Activity Plans Editor: to specify and validate Activity Plans,
    - Individual Activity Editor: to specify individual Activities taking advantage of the information provided by the 3D-RVP,
    - Annotations Editor: to specify and manage the annotations,
    - Paths Editor: to define paths; an important aspect here is the analysis of the defined path in terms of feasibility at each point, estimation of resources consumption as well as estimation of the traveled safety corridor.
    - Situational Awareness tools: a multitude of situational awareness tools are required ranging from the basic distance and slope measurement between two points to the rocks and directional slope detection and navigability/reachability/energy overlays.

3DROV Simulator

The 3DROV simulator includes:

- **Generic end-to-end simulation capabilities** in closed loop with the environment: To this end the simulator considers models ranging from motors and sensors to the atmospheric conditions in which they operate and to the software capable to run the operations. The simulator therefore considers the following three groups of models:
    - **Robotic System models**: it includes models of the motors, the sensors, the mechanical dynamics, and models of the power, the thermal, the communications, the data handling and the payloads subsystems.
    - **Environment Models**: it includes the Orbital and Timekeeping model to compute the significant ephemerides, the Atmosphere model to provide temperatures and solar fluxes at the operations location and the Dynamic Environment model to provide the dynamics of the robot/environment interaction.
    - **On-board Controller Model**: it includes a functional equivalent of the robotic system o/b controller based on the MUROCO activity which results are applied in the ExoMars MMS o/b software design.
    - **Ground Segment**: provides specific robotics MMIs (3DROCS) to perform the simulation.

- **Ability to attach onboard algorithms for testing**: During the development of the AIPlan4EU activity, specific planning technologies shall be validated. 3DROV provides the possibility to introduce algorithms to be tested into the o/b controller and the use of the environment model to provide the necessary inputs.
- **Different levels of fidelity for multi-domain physical system models**: to satisfy the different simulation objectives it disposes of models with different levels of details but also a simulator with configurable set of models to be included.

3DROV has been instantiated in several space robotics activities such as:

- The MSSTM Activity of the Mars Sample Return program,
- The ExoMars program for the ExoMars Operational simulator,
- The Analog-1 Moon exploration field-testing activity.

Requirements

Operations preparation

SP-010 – Automatic Activity Plan generation	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the automatic generation of Activity Plans		
Comment: Activity Plans are used for the tactical planning of the operations		

SP-020 - Activity Plan validation	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the validation of an Activity Plan; in case the Activity Plan is not valid the reason shall be reported		
Comment: This is needed to support a mixed initiative approach for planning		

SP-030 – Activity resources	Type: F	Verif: **T**
The UP shall be able to consider a) the resources consumption of the Activities, i.e., the duration, power consumption and memory mass generation and b) the resources availability		
Comment: The generated Activity Plans shall not consume more than the available resources		

SP-040 – External simulators use during planning	Type: **F**	Verif: **T**
The UP, during planning, shall be able to use external simulators to better estimate calculated resources		
Comment: There are Activities (e.g., warm-up) resources are not fixed, but can be estimated by external simulators		

SP-050 – Optimal plans	Type: **F**	Verif: **T**

	Optionally, the generated Activity Plans may be optimal wrt selected resources (duration, power, memory mass)
	Comment: Optimizing the resources is interesting in order to increase the science return

SP-060 – Plans consolidation	Type: **F**	Verif: **T**
The UP shall be able to consolidate several ‘partial’ Activity Plans to a single ‘consolidated’ Activity Plan		
Comment: Partial Activity Plans are generated by different science teams while the final ‘consolidated’ plan is generated by the engineering operations team		

SP-070 – Use of existing Activity models and associated tools	Type: **D**	Verif: **R**
The Space Use Case TSB shall be compliant with the existing Activities modeling approach and the associated tools (3DROCS)		
Comment: The objective is to add the UP functionality on existing operational tools		

SP-080 – Use of web-based interfaces	Type: **I**	Verif: **R**
The UP shall provide services based on web-based interfaces		
Comment: Operations planning is a geographically distributed process		

SP-090 – It shall be faster to automatically generate Plans than manually	Type: **P**	Verif: **T**
It shall be at least 20% faster to automatically generate Plans than manually		
Comment: During planetary exploration operations the available time for the ‘planning phase’ is restricted to few hours		

SP-100 – Impact of the UP in the Space domain	Type: **Impact**	Verif: **R**
The UP allows to decrease the duration of the operations planning cycle increasing so the exploitation of the communications window with the exploration robotics assets		
Comment: The time between two communication windows is very short; automatic planning allows to exploit this time in a more efficient way		

T2.2 Agriculture domain

Context

World-wide, agriculture faces the need to produce food for a growing population. In Europe with its high and rising standards for environmental protection, this has produced an immense pressure on the efficiency of agricultural production. In reaction to that, agriculture tech has become a highly digitized, high-technology branch involving powerful and expensive machinery. This further intensifies the pressure on farmers and contractors to care about effective and efficient processes with their capital-intensive property (both machinery and soil), meeting both economic and ecological constraints for a sustainable business model. In that situation, automatization, optimization and planning methods are natural choices, and they can build on existing high levels of digitization at modern farms and contractors.
One problem class is campaign planning for harvesting arable crops. It consists of various coordination and optimization problems on several levels of detail. In general, it deals with the problem of coordinating the available machines in their joint farming processes, taking additional constraints and optimization criteria into account. Campaign planning is needed in particular in harvesting processes, which involve harvesters, overloading and transport vehicles, and possibly other resources and agents, depending on the type of crop.

The harvesting processes for the various crops, e.g., silage maize, wheat, forage or sugar beet, vary due to different general requirements and machinery. A particularly complex process is *silage maize harvesting*, which is therefore the focus of this use-case. The chopping, transport and compaction of the crop in the silo must be coordinated so that the machines are used to full capacity and downtimes are avoided. Efficiency is important here, as there are requirements in terms of time, economy and sustainability: When the crop is ripe there is only a limited time window to harvest it. Here, weather conditions can lead to further restrictions or deadlines. Efficiency is also important from an economic point of view, as the machines and labor are expensive and limited. Finally, sustainability aspects such as a reduction in fuel consumption or soil compaction risks could also play a role.

Planning Application

Campaign planning deals with the coordination of different machines and resources in their joint farming process.
In our particular use case of silage maize the harvesting of multiple fields in a given time frame and the storing of the maize in a silo need to be coordinated. This problem can be addressed in multiple levels of abstraction: On the highest abstraction level, the order in which the fields are harvested with a set of machines can be planned while taking into account properties like the positions of the fields, which result in different travel times, the ripeness of the crops, weather conditions and forecast, working hours, resources and the process of accumulating and compacting the maize in the silo. Also, the composition of the vehicle fleet could be planned and **optimized**.
But also more detailed subparts of the campaign planning problem present planning problems on their own. It is important to find a good level of granularity that is worth considering. One example is the **infield planning problem**. It deals with the coordination of the different machines on the field by planning feasible routes and coordination points given the field's geometry. While at least one harvester is harvesting the field, unloading vehicles must take over the crop and transfer it to a designated deposit point, such as a silo by the farm. Maize forage harvesters normally do not have a bunker. Therefore, loading vehicles must constantly drive next to them and take over the crop. It must be ensured that this is always possible and that the unloading vehicles can reach the harvester at the appropriate time on a feasible path. As the unloading vehicle must not drive on unharvested areas of the field, the drivable area constantly changes. This infield planning problem can vary a lot depending on the type of crop and the associated machinery. For example, in wheat harvesting the machines do have a bunker and therefore the transport vehicles only need to take over once the harvester's bunker is full.

Campaign planning and infield coordination both require continuous **monitoring and adaptation** of the plans and their execution. The volume of the harvested maize, and therefore the time until filling the load capacity of a particular vehicle, varies depending on the quality of the plants (e.g., moisture, average mass), the crop density on the field (which varies across different patches on the same field), and other parameters. Prior to the physical harvest, this can be estimated from data and experience; but any planning regarding the overloading process in space and time needs to be adapted to the actual conditions at execution time. Furthermore, in infield planning, there are not only deviations in the travel times of the loading vehicles, but also differences between the estimated and actual yield and therefore in the loading status of the harvesters and/or overloading vehicles. On a larger scale, the status of the harvesting processes on other fields would have to play a role if they simultaneously supply the same silo, as stocking forage in a silo requires a process of targeted compaction of the material. This compaction process is another important variable that can often lead to idle times and needs to be considered as well.


Harvesting fields by day and night. The crop is loaded directly onto the transport vehicle. Left image courtesy of CLAAS KGaA mbH, right image courtesy of Maschinenfabrik Bernhard Krone GmbH & Co. KG.

Silage maize harvesting

In our particular use-case we will focus on the overall silage maize harvesting coordination problem that is defined as follows. The description of the harvesting process chain is largely based on the formalization by T. Steckel[2].

A farmer grows maize on multiple fields. When the maize is ripe, it is transported to a silo and compacted. The maize is preserved by compaction in the silo, and the resulting silage is used to feed animals or in a biogas plant.

Already at the time of sowing, the farmer plans and influences the expected harvest date by selecting the corn variety and the sowing date. Often multiple varieties are sown to reduce the risk of a poor harvest. However, ultimately the final harvest date depends to a large extent on the climate and weather. In the expected harvest period, the farmer measures the ripeness of the maize and finally determines the date of harvest. The closer the harvest date, the better the farmer can estimate the expected yield. All fields are harvested directly one after the other and the crop is stored in the silo. The whole process takes between one and a few days, depending on the number of fields. After five days at the latest, the silo must be covered.

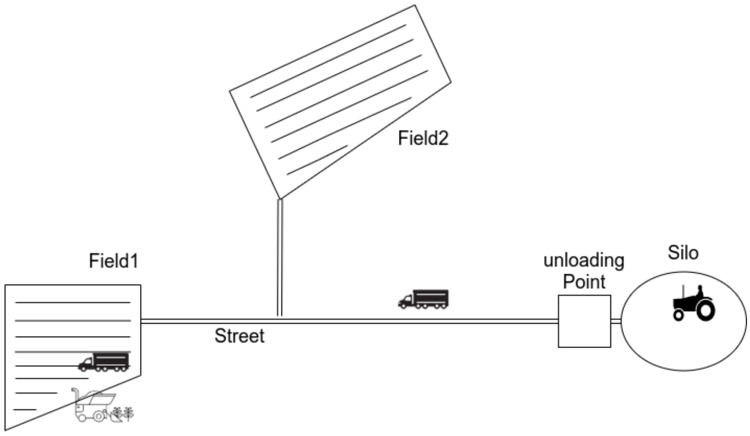
The harvest itself is often done by contractors because the farmer does not have all of the necessary expensive harvesting machinery. Together they choose the concrete machines that will be used and the order in which the fields will be harvested.

The farmer has to decide in which order the fields are harvested. That sequence could still be changed later on in the process. There are many different requirements that need to be considered here. For example, large fields should be selected for the night, and they should be scheduled to begin harvesting while it is still light. It should be avoided to harvest near residential areas during evening hours. In addition to that, sometimes the weather conditions play a role

[2] T. Steckel, 'Entwicklung einer kontextbasierten Systemarchitektur zur Verbesserung des kooperativen Einsatzes mobiler Arbeitsmaschinen', Universität Hohenheim, Institut für Agrartechnik, Hohenheim, Germany, 2018.

and as a rule of thumb the wettest field should be harvested during the day and possibly at the latest because transport vehicles might even get stuck on the field.
The silage maize harvesting campaign itself consists of three interdependent process steps:

1. Harvesting the maize on the field
2. Transport to the silo
3. Compaction in the silo



Harvesting the maize on the field can be referred to as the previously mentioned infield planning problem. At the beginning, the forage harvester normally cuts along the outer edges of the field to create a headland. This headland can then be used for turning maneuvers and the transport vehicles can traverse it. Afterwards, the field is harvested in rows. Due to the fact that maize forage harvesters do not have a bunker, a transport vehicle must constantly drive next to it. As soon as the capacity of the overloading vehicle is reached, another vehicle must take over immediately. This is called a coordination point. Otherwise, the harvester would have to stop and wait for the next one, which directly extends the total required time of harvesting the field.
Some domain-specific approaches focus exclusively on this infield planning problem for certain plants to coordinate the harvesters and transport vehicles on the field. They consist of a geometric preprocessing of the field, graph-search techniques for finding an optimal route, and a postprocessing step that aligns turns to the machines' kinematics. In our use case we will abstract away the infield planning by relying on such state of the art tools to infer expected coordination points in time and space.

Changing transport vehicles. Image courtesy of CLAAS KGaA mbH.

Transporting the crop to the silo. Images courtesy of Maschinenfabrik Bernhard Krone GmbH & Co. KG.

When the trailer of the overloading vehicle is filled, it **transports the crop** to the location of the silo. Naturally, this process step depends on the available routes between the fields and the silo as well as the routes' properties. The transporters need to adjust to the requirements and restrictions of the environment. These include speed limits of roads and the drivability of field paths due to humidity, but also residential areas or local events that can influence the choice of routes or the driving speed. To prevent the large machines from getting in each other's way on narrow roads, routes are often chosen so that narrow roads are only travelled in one direction.
When the transporter arrives at the silo, it is often first weighed on a scale. It then unloads the harvest in front of the silo. Afterwards, the transporter drives back to the field. The exact place of deposit is specified by the driver of the compaction vehicle, which then pushes it onto the silo. Here, it is important that the amount of yield that arrives at the silo can actually be processed by the compaction vehicles. Otherwise it would lead to waiting times for the transport vehicles, which in turn could ultimately lead to waiting times for the chopper.
The **compaction of the crop in the silo** is a critical process step. In order to make the silage durable, a sufficient density must be achieved. Studies have shown that silos are often not sufficiently compacted. Tractors equipped with shovels and additional weights usually serve as compaction vehicles. After pushing the yield onto the silo, they compact it by driving over it several times until the required density is achieved for the new layer. A compaction vehicle is able to compact a yield of three to four times its weight per hour.

The properties of all the machines of this process chain need to be well matched to each other. For example, the harvesting capacity of the forage harvester has a direct influence on the required capacities of the transport vehicles and the number of machines for compaction in the silo.

The harvesting process can be modeled in the light of various optimization criteria. These include quality, time and money. Furthermore, these differ from the perspective of the farmer or the contractor. The value of the quality of the silage is difficult to quantify. The farmer usually wants to achieve a minimum quality. Therefore, for the farmer, the overall goal of this use case is to optimize the harvesting process to reduce costs while achieving a desired quality of silage. Costs are ultimately measured in terms of money that the farmer must pay the contractor to harvest all of his fields. This again depends on the contractor's pricing models. Alternatively, the process could also be optimized from the perspective of the **contractor**, for whom the overall required time and the use of the available resources might be of higher importance. In this use case we will take the latter approach and model the process from the contractors point of view. This has the advantage that it provides additional decisions and options that can be considered for optimization. A contractor can harvest the fields of several farmers at the same time and tries to use his existing fleet of vehicles effectively. This increases the overall planning problem, including the selection of machines for the individual harvesting sub-problems. Furthermore, in case of contingencies it might allow shifting resources from one harvesting operation to another. For example, if it turns out during the harvest that the number of assigned transport vehicles is not sufficient it could be possible to shift transport vehicles from another harvest.

An initial plan will be created several days or even weeks before the harvesting date in order to find an expected optimal composition of the machines used and an ordering in which the fields are harvested. At the day of the harvest this plan will be refined in the light of the actual situation. Since there are also many contingencies during harvesting, such as deviations between the actual and the expected yields, it is important that the plan is constantly monitored and adjusted. At the same time, however, these changes also mean a disturbance to the process. Therefore, such changes should be kept to a minimum.

Pushing the crop onto the silo and compacting it. Image courtesy of CLAAS KGaA mbH.

Introductory Example

For a minimal example consider a farmer who grows silage maize on four fields. The silage maize needs to be harvested and stored in a silo.
Days or weeks before the anticipated harvest date the farmer measures the ripeness of the maize and sets a harvest date based on the weather forecast. The contractor and the harvester select the machinery and an ordering in which the fields will be harvested.

Input data:

- general problem data
    - fuel costs
    - labor costs
- possible machinery
    - forage harvesters:
        - harvesting performance
        - fuel consumption, consisting of
            - fuel consumption for chopping (depends on throughput)
            - fuel consumption for waiting
        - machine hourly rate
    - transporters

        - capacity
        - fuel consumption
            - for driving (depends on speed and load)
            - for waiting
        - max driving speed
        - unloading speed
        - machine hourly rate
    - compaction vehicles
        - weight
        - compaction performance
        - fuel consumption
            - for waiting
            - for compacting
        - machine hourly rate
- fields
    - size
    - expected yield
    - position
    - geometry
    - field access points
- silo
    - position
    - max capacity
- routes
    - either as a graph or a list of routes with start, end, distance, speed limit

Output:
- selection of machines and configurations, e.g.,
    - 1 forage harvester CLAAS Jaguar 950 with corn header ORBIS 750
    - 5 transport vehicles Steyr 6125 A with 2 2-axle tippers with capacity of 12 t
    - 2 compaction vehicles CLAAS XERION 4500 with additional 2 t front weight
- ordering of fields

At the **day of harvest** the above plan is checked and possibly refined.
Output:
- machine configurations (see above)
- activities and coordination points:

Time	Field	Activity
t1	field1	start overloading to transport vehicle 1(tv1)
t2	field1	end overloading tv1, tv1 drives to the silo with 40 km/h
t2	field1	start overloading to tv2
t3	silo1	tv1 unloads 24 t at silo

t4	silo1	compaction vehicle 1 (cv1) pushes 10 t maize onto the silo
t4	silo1	tv1 drives to field1
t5	field1	tv1 arrives at field1
t6	field1	end overloading to tv2, tv2 drives to silo1
t6	field1	start overloading to tv1
...	...	...
t100	field1	forage harvester (fh1) drives to field2
t101	silo1	tv1 drives to field2, on route7, with average speed of 40 km/h
t102	field2	fh1 starts harvesting field2, overloads to tv1
...	...	...

During the harvest the plan is monitored for consistency.
For example, the actual yield is higher than anticipated. Therefore, TF1 is full 2 minutes earlier than expected and TF2 needs to arrive at the field and take over at this time already.

Impact

Business Impacts

Current agricultural machines have a large nominal work output. However, the contingencies and the complexity of coordinating the harvest lead to the machines not being fully utilized. For example, they have to wait for others. In addition, the harvesting machines are expensive, and labor and fuel costs are also high. Optimizing the process and improving machine utilization would result in a reduction in harvesting costs and thus ultimately also in unit costs. This benefits both the farmer and the contractor. For the contractor, the utilization of his entire fleet could also be improved and his work capacity increased.

Other Impacts

Optimisation of the harvesting process also includes consideration of fuel consumption. Avoiding waiting times of the machines, adjusting driving speeds, or, if appropriate, selecting more fuel-efficient machines could reduce fuel consumption. This reduction can either be part of the overall process costs that are minimized or, if desired, it could also be the optimisation goal itself.
Other criteria such as the reduction of soil compaction could also be taken into account. Furthermore, optimizing the process at the silo can also lead to qualitative improvements such as the quality of compaction in the silo, which makes the silage more usable for the cows.

1 Measures of Success

As described in the following section the planning system will be evaluated based on recorded and simulated data. The main optimization metric is the overall cost of the harvest. Other parameters such as fuel consumption of all machines can also be compared.

In addition to the quality of the plan, the runtime for planning, monitoring and for adapting the plan need to be considered. It is crucial that the planning itself does not lead to delays. Here the initial planning could be done in advance, thus allowing several minutes or even a few hours. On the other hand, monitoring and plan adaptation should be fast enough in order to react appropriately.

2 Planning Integration

As described in Sections 2 and 3, due to natural contingencies in the harvesting process, monitoring and refinement or adaptation of the plan is key. An initial plan will be generated several weeks or days in advance in order to select the machinery and create a preliminary schedule. At the day of the harvest planning will be triggered again in order to generate a plan that adheres to given circumstances at the day of harvest. Finally, during the harvesting process the plan has to be monitored. It needs to be checked if the coordination points in the schedule are still feasible or if the plan needs to be adapted. It may even be possible to make ad-hoc improvements to the existing plan.

It will not be possible in the scope of this project to use the system already for the coordination of a live harvest. One reason for this is that this would require dedicated test fields, for which no resources are available in the project. Secondly, robust integration on real agricultural machinery is very complex. Therefore, the evaluation will be based on recorded data and a simulation. In consultation with a machine manufacturer, we will receive recorded data from a real harvest observed in the first year of the project in northern Germany. The contractor who carried out the harvest provides us with recorded data on routes and yields. This data will be used as a basis for the evaluation.

This will be used as input in order to determine the initial planning problem and to generate the initial plan. At that point, we can already compare whether our planning system would have come to a similar plan and how this would influence the overall cost. Furthermore, this data will also be used as a reference for occurring contingencies in the execution. This way it helps to parametrize a simulation tool that will be used to rest evaluate the planning system for dealing with deviations from the anticipated action steps and unexpected events.

Requirements

Operations preparation

AG-010 – Automatic Plan generation	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the automatic generation of plans fulfilling the given domain and planning constraints and goals.		
Comment: Activity Plans are used to command the campaign machines (agents) to timely perform the given actions to reach the final goal: harvest all fields and store all harvested yield in the available silos.		

AG-020 - Plan validation	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the validation of a plan; in case the plan is not valid the reason shall be reported		
Comment: To support a mixed initiative approach for planning.		

AG-030 – Manual plan generation	Type: **F/DE**	Verif: **T**
The UP shall provide the methods to manually create valid and invalid plans using the defined problem actions.		

	Comment: To facilitate validation of plans. The user should be able to easily generate plans using the defined actions, objects/fluents and initial states.

AG-040 – Plan inspection and simulation		Type: **F/DE/V**	Verif: **T**
	Given a plan and problem definition, the UP provides the methods to obtain the values of the state variables during each planned action or at a specific timestamp. For this, UP uses the same actions' definitions used for the definition of the problem.		
	Comment: For validation and analysis. In temporal planning, it should be possible to ask UP what is the state at a given timestamp, for example to test replanning. It should also be possible to simulate the execution of a plan and check/display the changes in the problem variables. To obtain the state values, the UP should use the same action definitions, i.e., the user should not have to create extra action simulators that do what the actions do.		

AG-050 – Management of problem and objects' parameters		Type: **F/DE**	Verif: **R**
	The UP shall allow the proper access and management of problem and objects' parameters via variables and constants. Supported types must include boolean and numeric (integers and reals). Supported types should include UserTypes and inheritance.		
	Comment: Problem and object parameters must be accessed and managed during planning. These include static/constant machine specifications (e.g., speeds, capacities, dimensions, id), machine dynamic/variable parameters (e.g., location, load), silo parameters (e.g., total capacity; silo unloading points' total and current capacity), field parameters (e.g., id, average yield mass per area unit), and transit parameters (e.g., travel distance between locations). Support of UserTypes and inheritance facilitates the problem definition and makes it cleaner.		

AG-60 – Usage of external functions during planning (effects)		Type: **F/DE**	Verif: **T**
	The UP shall allow the integration of external functions and use them dynamically during planning to compute corresponding action effects.		
	Comment: Some action effects depend on a) operations that are too complex to perform with the available UP operators; and/or b) access to external problem data. For instance, there are effects that depend on the computation of in-field routes based on previously computed routes and field geometries.		

AG-070 – Planning based on optimization criteria (optional)		Type: **F**	Verif: **T**
	Optionally, the UP supports optimization goals and plan accordingly.		

	Comment: In many cases a plan that satisfies the problem constraints is not enough; the plans must be optimal or near to optimal. The main goal of the agriculture use-case is to minimize/reduce the harvesting campaign duration, which will reduce the resource usage and therefore the campaign costs.

AG-080 – Planning time		Type: **F/Q**	Verif: **A**
	The time needed by UP to plan a harvesting campaign must be in line with the complexity of the campaign.		
	Comment: High (and indefinite) planning times are undesired. Since the real campaign is expected to have deviations from the plan, replanning is needed. The machines must not have idle periods caused by high (re-) planning times.		

AG-090 – Resource management - availability		Type: **F**	Verif: **T**
	The problem definition must consider the temporal availability of the resources and integrate it in the generation of plans.		
	Comment: The planner must not generate temporally-overlapping actions for a resource that make said resource unavailable. For instance, do not send a transport vehicle to a field while it is still busy unloading.		

AG-100 – Resource management - capacities		Type: **F**	Verif: **T**
	The problem definition must consider the temporal, dynamic storage capacities of the resources when generating plans.		
	Comment: The total capacity of the silos, silo unloading points, and transport vehicles must not be exceeded.		

Operations execution

AG-110 – Re-planning or plan repair		Type: **F/Q**	Verif: **T**
	When simulating the execution of a plan, the TSB and UP shall provide functionalities to update the plan based on the current state of execution.		
	Comment: Harvesting operations have many sources for deviations in the real world from the plans, e.g., due to traffic or differences between the actual amount of yield on a field from the estimated amount. We want to react to such deviations by re-planning.		

AG-120 – Plan simulation		Type: **F/Q**	Verif: **T**
	The TSB shall provide the tools to simulate the plans obtained from UP. The simulation might include deviations from planning parameters (e.g., amount of yield and transit durations). Results of the simulation (including deviations from the plan) can be used for replanning.		
	Comment: This is important for testing the robustness of the plan and to test re-planning.		

AG-130 – Plan visualization (optional)		Type: **F/Q**	Verif: **T**
	The TSB might provide the tools to visualize the campaign plans obtained from UP.		
	Comment: This is important for showing the plan to the user and for checking the plan.		

T2.3 Flexible manufacturing

Meritor HVS AB in Lindesberg manufactures rear and front axles for heavy vehicle applications. The plant contains both machining and assembly. Order horizon for assembly is short with all axles assembled marked with customer chassis number and all axles are delivered to customer in sequence (Make-to-order). The assembly process is also quite short with fast lead-times, which allows the planning of the assembly process to more or less mimic the customer needs.

The machining department has longer lead times, which forces us to look at order forecasts for planning. This, together with a complex machine park with numerous different process steps and different machines with different cycle times depending on the part produced, makes planning difficult.

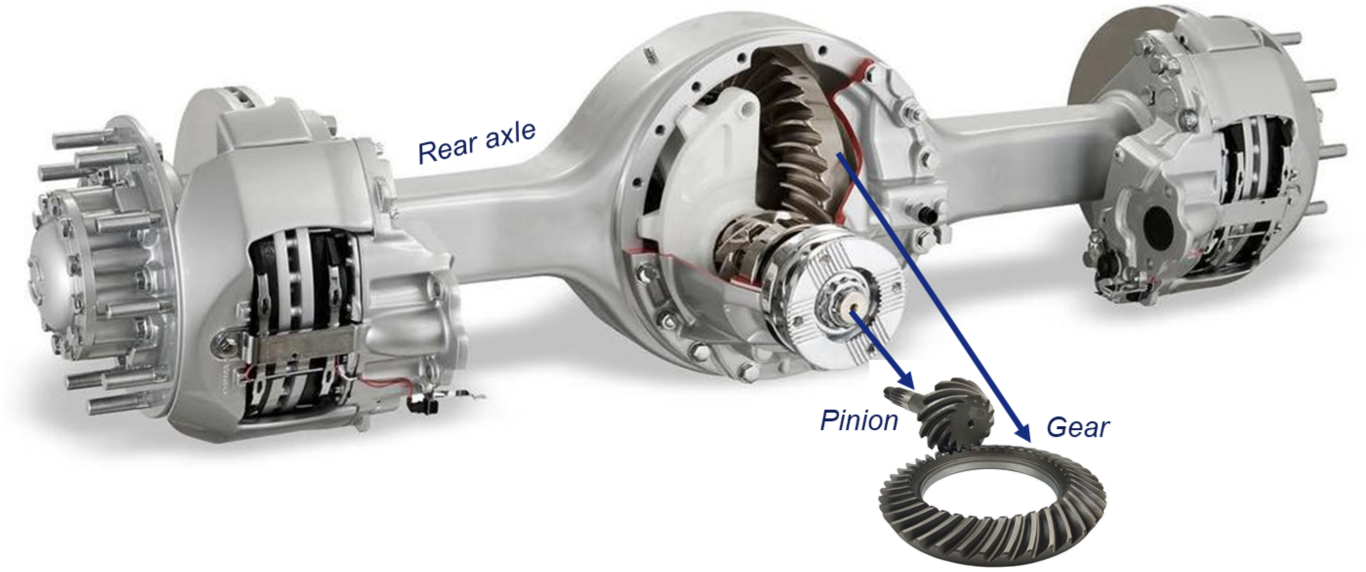
In this use case, we consider a component group named gearsets (Bevel gear & pinion), which has the longest lead time and involves multiple departments and processes to finish the product. Today, a human planner plans day by day, using data of parts that are available and parts that are required for production. The human planner attempts to create a production plan based on what he/she knows at the time. However, this is a very time-consuming process, and the result is almost always “wrong, as changes have occurred during the planning process itself. For example, machines can break down, target cycle times are not achieved, etc. Because of this, we currently use sub-planning activities in all machining departments that focus only on their own department. This leads to plans that are un-optimized for the plant.

We expect AIPlan4EU technology to provide the means to create a planning tool that will provide decision options for the central human planners to take the decision in real-time, based on accurate data for controlling the production.

Context

Meritor Lindesberg site produces rear and front axle assembly for commercial vehicles. The factory runs 24 hours and 5 days a week. Major parts of manufacturing processes are covered under the manufacturing execution system (MES). Almost all the machining cell component handling is automated. In some parts of the factory pallets/ kits handling and transportation are managed by automated guided vehicles (AGV).
Bevel gear and pinion set for the rear axle is an in-house produced and highly engineered component. Both bevel gear & pinion undergo similar process steps but in different machining cell-like soft machining, tooth cutting, heat treatment, hard turning, and finally becomes a set in the lapping cell.
Inhouse, we manufacture many gearing parts for rear axles like bevel gear & pinion, planetary gears, differential gear sets, cylindrical gears, ring gears but bevel gear & pinion has the longest lead time which decides the axle manufacturing lead time. On the other side, both components bevel gear & pinion undergo multiple departments to convert the input material to the final product, and they become a set after lapping operation. Planning & scheduling of bevel gear & pinion at different stages is highly complex and completely managed by excel spreadsheets. There are many aspects that make planning & production of bevel gear & pinion more complex, like:

- High variants of gear sets under each product family
- Make to stock for better capacity utilization
- Complex setup process from one part to another
- Availability of right quantity of match gear and pinion before lapping process to convert the final product

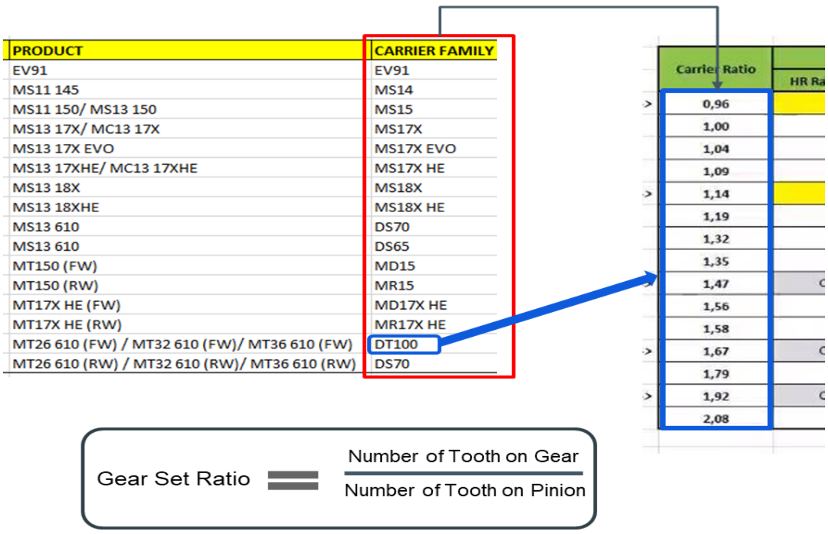


Gear set manufacturing process

Bevel gear and pinion undergo similar operation individually and become a set after lapping operation. After lapping they must go together in a set for the carrier assembly.

High Variant under each family

In the picture below, we can see the level of complexity in gear and pinion planning. For each carrier family, we have a 10+ ratio of gear and pinion. Gear set ratio means the number of teeth on gear divided by the number of teeth on the pinion. For every ratio in a carrier family gear and pinion having a unique part number.



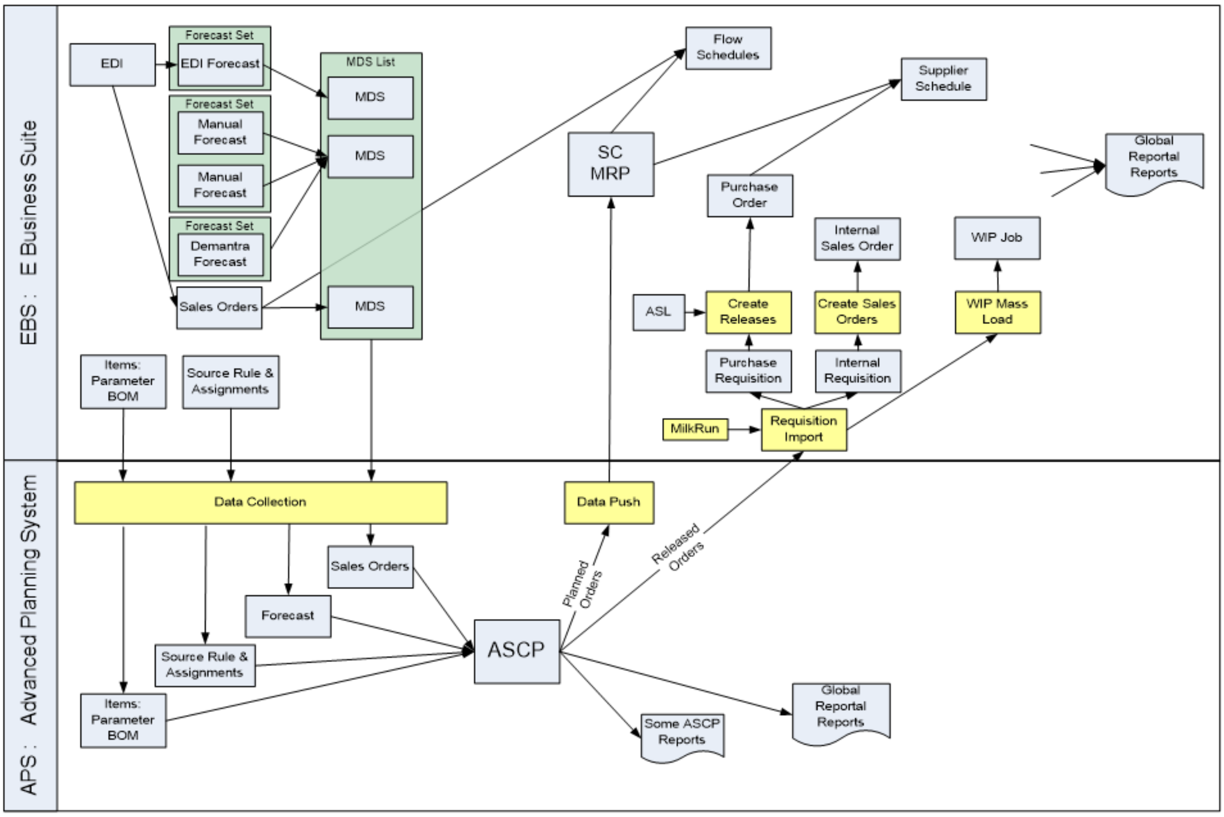
Drawbacks of the current solution

- When gearsets are produced the gearset might not be needed any more, a fact which may become known with a lead time of 14 days.
- Difficult to accommodate sudden changes in the plan to serve short-lead-time customers. This is typically done by "consuming" parts planned for a long-lead-time customer hence the contingency implies the need to re-plan their production.
- A new person takes years to understand the planning process and it's very difficult to find a replacement for a human planner, internally or externally.

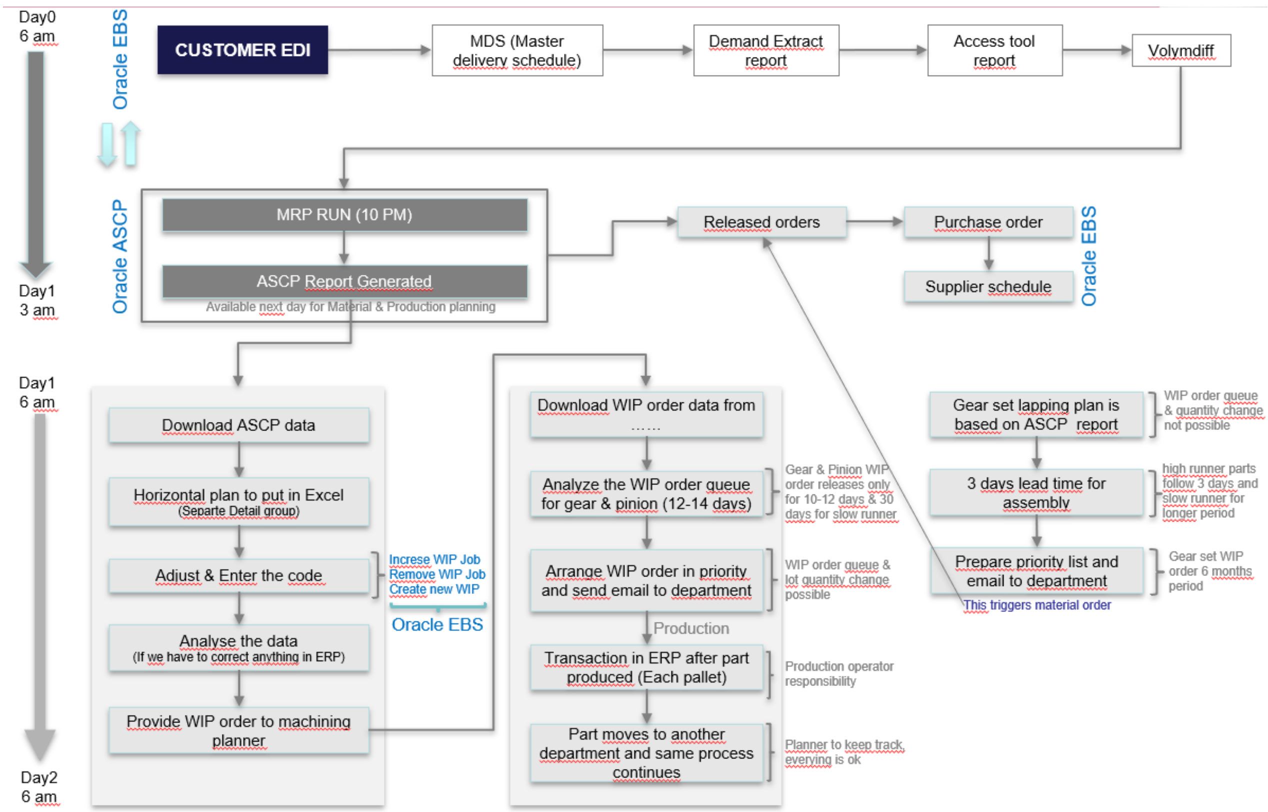
Planning Application

The Meritor use case is focused on flexible manufacturing planning capacities to improve the general workflow of machining human planners specific to bevel gear and pinion. Today's solution is completely manual and managed by multiple tools like ERP (Enterprise resource planning), WMS (Warehouse management system), ASCP (Advance supply chain planning) planning engine, Excel spreadsheets, etc. and several different human planners are involved from customer contact, material planner, machining planner and department planner. As a result, planning & scheduling of bevel gear and pinion becomes a highly complex manual process. Today, the human planner's role becomes very important in the overall process for efficient planning based on knowledge and available data. The human planner makes decisions based on his/her visibility and facts available at that point in time. The planning is based on the customer order and then translated to make-to-stock. Make-to-stock means that we aim to keep a fixed stock level between the machining and assembly department. At the same time, we also try to reduce the stock level keeping the right/ required product mix in the stock.

The schematic chart below shows the end-to-end planning process. EBS (E Business suite) & APS (Advanced planning system) are the two different ERP (Oracle) environments. EDI forecast, manual forecast, flow scheduling, supplier scheduling are processed under EBS environment , whereas Data collection related to forecast and supply chain planning report is generated by APS environment.

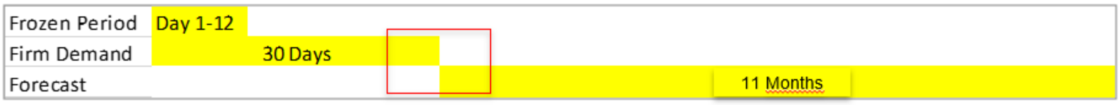


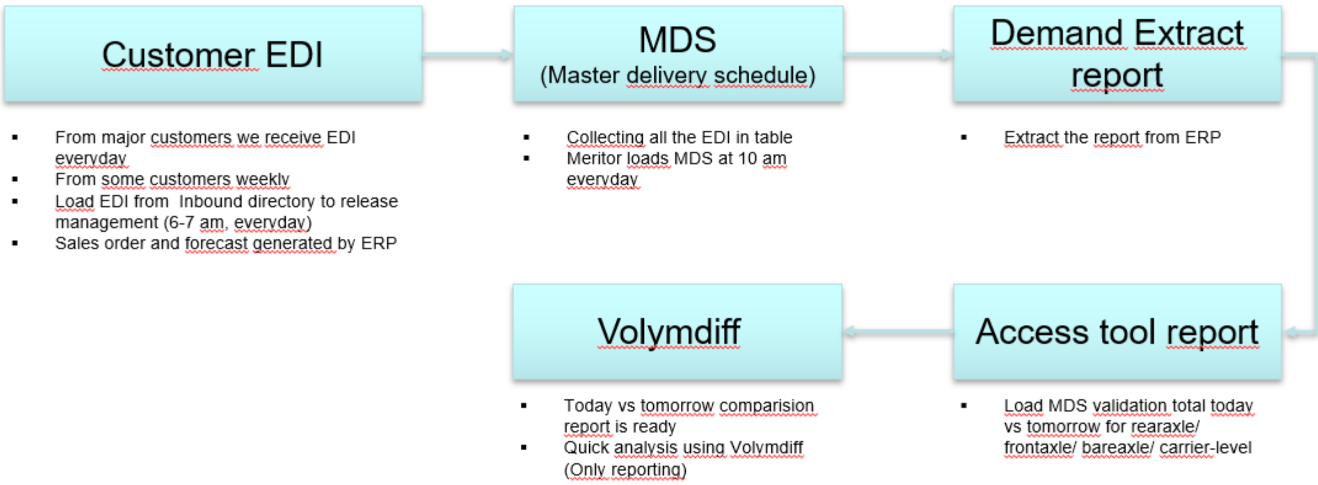
The chart below shows the current planning process starting from customer order to supplier scheduling and production scheduling.



Customer Contact

A customer order called EDI (Electronic data interchange) is received from each customer directly in our ERP (Oracle EBS). EDI orders are fully defined in terms of finished axle part numbers and are released for 12 months. In that 12 month of the forecast, the first 30 days are firmed order and out of 30 days 12 days are frozen orders (Frozen orders cannot be changed). A new EDI received every day will be reviewed by customer contact and compared against the previous day EDI order as stated in the below process flow. Demand extract report (commodity/ component level breakdown of EDI) will be generated from the ERP based on master demand schedule (compiled file with all EDI) and with the support of M.S Access tool, EDI comparison report (Volymdiff) will be generated for information and action purposes. The Volymdiff report generated from the M.S Access tool will only provide the information to customer contact to take necessary action in case of fluctuation between yesterday and today's demand. If we don't take any action to correct the EDI in the ERP, MRP (Material requirements planning) will autorun the same day in Oracle ASCP after 10 pm and will generate the report for material and production planners for next day planning. Material and production human planners use the ASCP report to send the material forecast to supplier and production planning. If any error was not addressed in the demand file, then it will lead to wrong material forecast to supplier and error in WIP order queue.





- If validation of MDS is accepted no action is required.
- If validation of MDS is not accepted we correct order/forecast and re load MDS and verify that MRP run ok next morning.
- If correction is not possible or MRP failed we inform Supply Chain and re load MDS and verify that MRP run ok during night
- If correction is not possible we inform Supply Chain.

Rear	Load Date	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2133	2134	2135	2136	2137	2138	2139	2140
Total	2021-05-17	1842	2338	2427	2437	2730	2761	2933	3126	2103	1431	3445	2762	2874	2793	3025	2570	2800	2878
Total	2021-05-18	1225	2330	2405	2440	2748	2793	2928	3136	2105	1434	3347	2776	2875	2796	3026	2567	2801	2879
Total diff	(inkl. BA)	-617	-8	-22	3	18	32	-5	10	2	3	-98	14	1	3	1	-3	1	1
Carrier	**Load Date**	**2120**	**2121**	**2122**	**2123**	**2124**	**2125**	**2126**	**2127**	**2128**	**2129**	**2130**	**2131**	**2132**	**2133**	**2134**	**2135**	**2136**	**2137**
Total	2021-05-17	496	389	393	357	380	311	471	440	436	493	18	16	18	447	332	451	342	388
Total	2021-05-18	497	389	388	361	380	311	473	440	434	493	16	22	37	463	353	473	374	421
Total diff		1	0	-5	4	0	0	2	0	-2	0	-2	6	19	16	21	22	32	33
Front	**Load Date**	**2120**	**2121**	**2122**	**2123**	**2124**	**2125**	**2126**	**2127**	**2128**	**2129**	**2133**	**2134**	**2135**	**2136**	**2137**	**2138**	**2139**	**2140**
Total	2021-05-17	1691	2096	2405	2277	2497	2348	2493	2482	1338	1420	2486	2619	2484	2624	2664	2276	2478	2563
Total	2021-05-18	1241	2096	2387	2274	2510	2367	2496	2497	1339	1425	2539	2615	2485	2627	2668	2273	2485	2568
Total diff		0	0	-18	-3	13	19	3	15	1	5	53	-4	1	3	4	-3	7	5
Bareaxle	**Load Date**	**2120**	**2121**	**2122**	**2123**	**2124**	**2125**	**2126**	**2127**	**2128**	**2129**	**2133**	**2134**	**2135**	**2136**	**2137**	**2138**	**2139**	**2140**
Total	2021-05-17	285	407	220	319	465	660	557	761	889	30	1142	294	483	292	371	298	591	509
Total	2021-05-18	145	399	219	329	464	665	559	761	889	30	1050	299	484	291	371	298	591	508
Total diff		-140	-8	-1	10	-1	5	2	0	0	0	-92	5	1	-1	0	0	0	-1

Example: We receive X Variant rear axle order of 100 qty from customer A. The X Variant rear axle is broken down to component/ commodity level by the Oracle ASCP tool. This is the base used by the WIP planner to create a work order.

WIP planners

WIP (work in process) planners release work orders based on customer EDI (ASCP report). These orders are known and can be modified for 120 days (6 months). After about 108 to 117 days, they typically become frozen and do not change anymore. A WIP planner analyzes the ASCP report and creates the work order for individual commodity and part number in the ERP tool (Oracle EBS). On one hand, the WIP order is a base for machining planners and on the other side, it triggers a new material order forecast for suppliers

Example: To produce 100 quantity X variant rear axle, we need 100 bevel gear and 100 pinions in a set. WIP planner receive the data from ASCP report and release to the ERP tool (Oracle EBS) to create the work order of 100 bevel gears and 100 pinions.

Material Planner

After WIP orders are released, the material planner sends the material forecast to the suppliers on daily basis. The planner is responsible for tracking the shipment from the supplier location to the Meritor unloading dock, the planner is also responsible for highlighting the critical shipments and sending the approval request for airlift.

Example: The material planner will release the material order consisting of 100 bevel gear forges and 100 pinion forges to the supplier. A forge is the input material for producing gears and pinions.

Machining area planner

The machining area planner runs the report for the overall machining department based on stock on hand and frozen customer orders (3-12 days). Based on this he/she creates a machining plan and sends it to the individual department planner through an email. The machining planner ensures that the right part number is in the right place at the right time.

Example: The material area planner will verify the availability of 100 gear and 100 pinion stocks and sends the machining plan to department planner to meet stock requirement for a frozen customer demand.

Department planner

Every department has their own department planner who adjusts the machining plan to match the machining cell capacity. The main purpose of this adjustment is to reduce the setup change and better utilization of capacity. The department planner orders material in the WMS system, which trigger material movement by AGV or manual forklift from storage to the machining cell or movement to next operation. The planner is also responsible to print the new label for finished components and reporting the finished work order in the ERP tool (Oracle EBS).

Example: Department planner executes the machining plan of 100 gear and 100 pinions in the most effective way. A different department planner continues the same process in their department until we finish all operations. Finally, we convert 100 gear and 100 pinions into 100 gearsets, which are ready for the assembly department to build 100 finished axles.

Key problems & Potential solution

The planning process today is a manual workflow and several human planners are involved from start to finish. We have identified several key issues that are expected to be addressed in the use case.

Identified key problems below:

1. Sometimes Interplant demand changes just before 2- or 3-days of delivery. When we accept any demand increase, we generally check the impacts in the frozen period (12 days). However, we need to check the impact outside the frozen period, which require more time and resources. Indeed, this requires to:

- Find the change and track the change
- See the impact of change
- Re-plan the production to accept the change

2. One forging of bevel gear or pinion is connected to many finish parts, if we consume more forgings for any demand increase, we need to know the impact. Forging shortage could happen since the same forging will be needed for another part number. The lead time of bevel gear & pinion forging is 5-6 months and that makes it very clear that the impact should be known before accepting the demand change.

3. If any machine is under major breakdown, we need to know the impact on assembly. Which machine is the best alternative to solve the situation, since some machines are very critical for certain part numbers? In this case, the production re-plan is very critical to avoid the losses in assembly and the losses in the machining area.

4. In some cases, delay in forgings from the suppliers leads to narrow planning due to priority but we miss knowing the impact on other deliveries.

5. In case of non-availability of the right forging, we produce any other part from stock to safeguard the SDLH and capacity. This leads to inventory build-up at each stage of gear and pinion flow and a shortage of forging because of fewer checks on the impact.

6. Declared maximum production output is based on a certain product mix, this needs to be taken care of in case of production scheduling and re-planning.

7. Based on the WIP queue, we move the forgings to the annealing operation and sometimes we see a reduction in demand or zero demand. This ends up with bad capacity utilization.

Potential solutions below:

- ➔ Today the planner uses several different systems and tools to perform the planning process e.g., ERP, WMS, spreadsheets, and different reports. A common "Tool" where planning is performed that combines the different data sources would be a huge benefit in many aspects such as time, quality, and accuracy
- ➔ Collecting data manually from the ASCP tool and compiling different reports are time-consuming tasks that are performed for each product group every day. Automating these tasks would be timesaving for the planning process.
- ➔ The current method for creating a WIP job/ work order in ERP (Oracle EBS) is completely manual and therefore also time-consuming.

  The future preferred process should be automated, where an automated planning tool will provide options for WIP order creation, and deletion and propose corrections where needed. Also, once the WIP planner approves the changes, the planning tool should automatically feed the information into the ERP tool for the creation of work orders.

- ➔ All data in the current plan is in many ways old since updates are performed once per day.

  One of the challenges that need to be addressed is how to provide updated data to the automated planning tool at planning time, and how to automatically trigger the appropriate re-planning operations when needed.

Relevant data for taking decisions that needs to be automated

A. Constant
   - Stock levels
   - Machine capacity and setup
   - Data from the ASCP tool
   - WIP order queue

B. Contingent:
   - Customer order EDI
   - Backlog of material due to machine breakdown etc.

Introductory Example

At any given time WIP Order (work order) queue is visible for 120 days based on the customer EDI. Meritor schedules the machining plan for gear and pinion 12 days in advance as per the frozen customer order which has a lead time of 3-12 days. Once the WIP planner receives the ASCP data, he/she will investigate the gross requirements and the new order will be adjusted in the WIP order queue. Since we produce the parts in the machining department in batches, (human) department planners always focus on machine utilization. It is also important to cluster jobs for the cutting machinery in such a way as to minimize setups. This is because each setup potentially diminishes the lifetime of the blade. This clustering occurs based on gear ratios. On the other hand, it is equally important that the right match of gear and pinion is available for the lapping process as per the plan. Today the human planner refers to multiple excel spreadsheets and tools for both planning and scheduling, and the quality of the plan depends entirely on the available data and knowledge/experience of the planner.

Here, we see an opportunity to introduce an AI planning tool, which can help to schedule the machining plan for better capacity utilization, keeping eye on the WIP order plan vs actual and main assembly requirements. For the success of such a tool, information related to machines like “Takt” (declared time for one machining job), setup time, and WIP-related real-time information will be key. Such a tool will reduce the multiple excel spreadsheets and the human planner will have the role of approval authority for the final plan prepared by the tool.

Needed features :

1. Present the impact for the frozen & forecast period before accepting the customer demand change
2. Production scheduling using the machine capability data, takt time/cycle time data, and the optimized model mix data for maximum capacity utilization
3. Human planners should have options to select and approve the production schedule
4. Could be able to see the impact on the assembly and customer delivery in case of a machine breakdown, material shortage, and major quality issues with produced gearset batch
5. In case of a machine breakdown, re-planned production options for the human planner to avoid capacity loss and delivery loss

Table 1:

ASCP Report: Week wise gross requirement against each part number

		Gross requirements		1059	1124	1321	2551	2931	2715	1973	2645	2847	3019	3212	3546	2611	3034	2591	4207	1053
		Projected available balance	2629	6898	7260	8168	8224	7960	8347	8784	9351	9748	10175	9682	9007	9025	9003	9402	8761	8082
		WIP		1059	1124	1321	2551	2931	2715	1973	2645	2847	3019	3212	3546	2611	3034	2591	4207	1053
		Working days		4	4	5	5	5	5	4	5	5	5	5	5	5	5	5	5	1
		Average/dag		265	281	264	510	586	543	493	529	569	604	642	709	522	607	518	841	1053
				WEEK13	WEEK14	WEEK15	WEEK16	WEEK17	WEEK18	WEEK19	WEEK20	WEEK21	WEEK22	WEEK23	WEEK24	WEEK25	WEEK26	WEEK27	WEEK28	WEEK29
HP:0:EU_ASCP Current Preference : Default				29-MAR-2	05-APR-20	12-APR-20	19-APR-20	26-APR-20	03-MAY-2	10-MAY-2	17-MAY-2	24-MAY-2	31-MAY-2	07-JUN-20	14-JUN-20	21-JUN-20	28-JUN-20	05-JUL-20	12-JUL-20	19-JUL-20
NA:LIN	A42538F	Gross requirements		1	5	10	30	37	37	34	71	39	32	29	34	61	55	60	71	37
NA:LIN	A42726	Gross requirements		0	0	0	2	6	14	0	9	0	5	5	4	0	2	2	8	0
NA:LIN	A42790	Gross requirements		0	3	5	1	6	11	3	1	3	4	4	8	3	6	5	12	0
NA:LIN	A43072	Gross requirements		18	0	8	11	21	4	2	2	20	18	12	20	11	14	12	32	1
NA:LIN	A43076	Gross requirements		0	15	1	3	3	5	3	3	0	0	0	0	0	7	4	14	0
NA:LIN	A43080	Gross requirements		2	3	14	1	1	5	1	1	1	3	0	0	1	1	3	1	1
NA:LIN	A43084	Gross requirements		0	0	20	7	8	9	6	7	10	10	9	11	7	0	0	3	0
NA:LIN	A43088	Gross requirements		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NA:LIN	A43096	Gross requirements		10	0	0	0	0	0	0	0	4	4	14	25	2	5	6	19	0
NA:LIN	A43100	Gross requirements		2	4	0	7	22	14	19	30	6	8	7	4	6	9	4	33	2
NA:LIN	A43104	Gross requirements		11	6	27	20	7	24	11	3	9	8	7	3	11	16	21	18	1
NA:LIN	A43448K4F	Gross requirements		3	7	8	30	47	34	33	58	45	49	51	49	48	51	25	40	24
NA:LIN	A43450K4F	Gross requirements		0	3	10	27	43	32	28	54	42	48	49	46	45	47	19	40	19
NA:LIN	A43456K4	Gross requirements		6	2	7	15	5	9	6	1	1	5	11	10	16	24	11	49	41
NA:LIN	A43458K4	Gross requirements		8	8	23	19	5	22	8	16	0	15	18	10	16	22	23	62	44
NA:LIN	A43460K4	Gross requirements		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NA:LIN	A43468K4	Gross requirements		4	0	0	6	0	8	6	4	2	0	4	2	6	0	6	0	10
NA:LIN	A43470K4	Gross requirements		4	0	0	4	0	8	6	0	0	0	2	0	0	0	2	0	0
NA:LIN	A45040	Gross requirements		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
NA:LIN	A45042	Gross requirements		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
NA:LIN	A45044	Gross requirements		0	0	2	1	9	2	4	5	2	1	2	3	7	5	1	0	1
NA:LIN	A45046	Gross requirements		0	0	5	1	5	5	0	0	2	0	2	3	0	3	3	4	2
NA:LIN	A45048	Gross requirements		0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
NA:LIN	A45166	Gross requirements		0	0	1	1	0	2	0	0	0	2	0	0	0	0	2	3	0
NA:LIN	A45168	Gross requirements		0	1	0	0	3	2	1	2	1	3	5	2	1	1	1	0	0
NA:LIN	A45170	Gross requirements		0	0	1	1	7	3	3	1	3	2	4	4	2	3	1	1	0
NA:LIN	A45172	Gross requirements		0	9	0	22	27	12	23	4	12	9	10	11	2	10	6	28	0
NA:LIN	A45174	Gross requirements		0	1	0	1	0	5	4	2	10	5	5	6	11	10	11	2	2
NA:LIN	A45180	Gross requirements		62	70	86	149	194	195	123	267	176	205	227	303	272	276	163	238	38
NA:LIN	A45182	Gross requirements		116	78	90	163	130	129	59	112	168	122	114	173	191	142	194	197	28
NA:LIN	A45182-C	Gross requirements		0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
NA:LIN	A45184	Gross requirements		11	8	20	29	72	49	29	51	54	27	38	54	55	34	21	26	7
NA:LIN	A45184-C	Gross requirements		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Table 2:

WIP Orders queue with batch operation start and end date

	DETGR_	ITEM_NUMBER	ITEM_DESC	FAMILI_CAT	WIP_ENTITY_NAME	OP_SEQ	QTY_IN	RES_CODE	RESOUR	CUM_C	WIP_STAR	OP_START	OP_END_DAT	DAYS_AHEAD	
90	331	42050K4	KRONHJUL 3.09 MD15 (	MD15	WIP46546453	10	232	GLODG	3	3	2021-03-05	2021-03-05	2021-03-10	-14	
52	331	45180K-5	KRONHJUL 17X HE 2.47	17XHE	WIP46488390	10	345/570	GLODG	3	3	2021-03-05	2021-03-05	2021-03-10	-14	24-mar
97	331	42042K4	KRONHJUL 3.09 MR/MS1	MR/MS15	WIP46546451	10	192	GLODG	3	3	2021-03-12	2021-03-12	2021-03-17	-7	24-mar
02	331	45566K-5	KRONHJUL 17X HE 2.31	17XHE	WIP46642704	10	493/733	GLODG	3	3	2021-03-12	2021-03-12	2021-03-17	-7	24-mar
99	332	43471K4	PINJONG DT100H 1.79	DT100H	WIP46546459	10	80	GLODG	3	3	2021-03-12	2021-03-12	2021-03-17	-7	
50	331	42050K4	KRONHJUL 3.09 MD15 (	MD15	WIP46642699	10	816	GLODG	3	3	2021-03-18	2021-03-18	2021-03-23	-1	
66	331	45186K-5	KRONHJUL 17X HE 3.08	17XHE	WIP46409465	10	156	GLODG	3	3	2021-03-18	2021-03-18	2021-03-23	-1	
87	331	45180K-5	KRONHJUL 17X HE 2.47	17XHE	WIP46727247	10	705/480	GLODG	3	3	2021-03-23	2021-03-23	2021-03-26	2	
84	331	45188K-5	KRONHJUL 17X HE 3.36	17XHE	WIP46727250	10	132	GLODG	3	3	2021-03-23	2021-03-23	2021-03-26	2	
89	331	42152K4	KRONHJUL 4.50 MR/MS1	MR/MS15	WIP46727244	10	32	GLODG	3	3	2021-03-24	2021-03-24	2021-03-29	5	
19	332	45379K-5	PINJONG 17X HE 2.47	17XHE	WIP46727261	10	735	GLODG	3	3	2021-03-25	2021-03-25	2021-03-30	6	
34	331	42538K4F	KRONHJUL 2.50 EV91	EV90/91	WIP46727246	10	88	GLODG	3	3	2021-03-26	2021-03-26	2021-03-31	7	
60	332	42601K4	PINJONG MR/MS15 5.67	MR/MS15	WIP46727256	10	38	GLODG	3	3	2021-03-26	2021-03-26	2021-03-31	7	
90	332	43255K3	PINJONG MS17X 5.63 B	MS17X	WIP46727252	10	65	GLODG	3	3	2021-03-26	2021-03-26	2021-03-31	7	
128	331	1522249	KRONHJUL DT100H/DS65	EV72/DT100H	WIP46727240	10	96/100	GLODG	3	3	2021-03-29	2021-03-29	2021-04-01	8	24-mar
218	331	42234K4	KRONHJUL 5.67 MD15 (	MD15	WIP46727245	10	36	GLODG	3	3	2021-03-29	2021-03-29	2021-04-01	8	
230	331	43104K3	KRONHJUL MS17X 5.63	MS17X	WIP46727241	10	24	GLODG	3	3	2021-03-29	2021-03-29	2021-04-01	8	
235	332	42449K4F	PINJONG 3.09 KULPENA	MT150E	WIP46727254	10	606	GLODG	3	3	2021-03-29	2021-03-29	2021-04-01	8	
240	332	42677K4	PINJONG 5.67 MD15 (P	MD15	WIP46727258	10	45	GLODG	3	3	2021-03-29	2021-03-29	2021-04-01	8	
295	331	45554K-5	KRONHJUL 17X HE 2.17	17XHE	WIP46727251	10	90	GLODG	3	3	2021-03-30	2021-03-30	2021-04-06	13	
330	332	42673K4	PINJONG 4.50 MD15 (P	MD15	WIP46727257	10	90	GLODG	3	3	2021-03-30	2021-03-30	2021-04-06	13	
395	332	42679K4	PINJONG 6.17 MD15 (P	MD15	WIP46727259	10	45	GLODG	3	3	2021-03-30	2021-03-30	2021-04-06	13	
465	332	45797K-5	PINJONG 17X HE 2.17	17XHE	WIP46727264	10	90	GLODG	3	3	2021-03-30	2021-03-30	2021-04-06	13	

Table 3:

Planning on hand and demand per item per day

MERITOR
Report id: 846

Green = Full Coverage
Yellow = If you order in time you can Cover demand
Red = No Coverage
Blue = Incoming ASN Shipment

Planning Onhand and Demand per Item
Run Date :Mar 25,2021

	Product Number	Planner Code	Item Type	Fixed Lead Time	Release Time	Days of Stock	Days of Safety Stock	Difference	On hand	WIP On Hand	Past Due	2021 3 25	26	27	28	29	30	31
71	1524907	330	LIN Subassembly	3	0	34	0	34	74	801	0	0	0	0	0	-10	-5	
72	1524908	330	LIN Subassembly	3	0	43	0	43	221	1501	-1	-1	0	0	0	-4	0	
73	1524909	330	LIN Subassembly	3	0	32	0	32	262	1390	-66	-3	-37	0	0	0	0	
74	1524910	330	LIN Subassembly	3	0	26	0	26	110	1385	-40	0	-26	0	0	0	0	
75	1524911	330	LIN Subassembly	3	0	32	0	32	88	966	0	-12	0	0	0	-4	0	
76	1524957	330	LIN Subassembly	3	0	54	0	54	148	584	0	-1	-1	0	0	0	-3	
119	1524938	330	LIN Subassembly	3	0	48	0	48	98	780	0	0	0	0	0	0	-1	
120	1524940	330	LIN Subassembly	3	0	25	0	25	101	1679	0	0	0	0	0	0	0	
121	A43468K4	330	LIN Subassembly	3	0	90	0	90	43	52	0	-4	0	0	0	0	0	
122	A43470K4	330	LIN Subassembly	3	0	999	0	999	76	0	0	0	0	0	0	0	0	
123	1524939	330	LIN Subassembly	3	0	46	0	46	119	791	0	0	0	0	0	0	-7	
124	1524941	330	LIN Subassembly	3	0	34	0	34	307	1483	0	-21	0	0	0	0	0	
125	1524942	330	LIN Subassembly	3	0	32	0	32	107	1258	0	0	0	0	0	0	0	
126	1524943	330	LIN Subassembly	3	0	32	0	32	100	879	-49	-17	-3	0	0	0	-4	
127	A43448K4F	330	LIN Subassembly	3	0	15	0	15	7	1022	0	0	0	0	0	0	-2	
128	A43450K4F	330	LIN Subassembly	3	0	34	0	34	59	970	-1	0	0	0	0	0	0	
129	A43456K4	330	LIN Subassembly	3	0	25	0	25	29	238	-3	-1	0	0	0	0	0	
130	A43458K4	330	LIN Subassembly	3	0	25	0	25	57	350	-11	-3	0	0	0	0	-2	
242	B-44016	330	LIN Subassembly	3	0	12	0	12	39	567	-10	-6	0	0	0	0	-10	
292	1523159	330	LIN Subassembly	3	0	26	0	26	0	0	0	0	0	0	0	0	0	
298	1524151	330	LIN Subassembly	3	0	33	0	33	79	866	0	0	0	0	0	0	-8	
301	1524252	330	LIN Subassembly	3	0	75	0	75	18	83	0	0	0	0	0	0	-1	
303	1524294	330	LIN Subassembly	3	0	999	0	999	27	0	0	0	0	0	0	0	-1	
443	A-44068	330	LIN Subassembly	3	0	43	0	43	140	1349	0	0	0	0	0	0	-2	
549	A42042	330	LIN Subassembly	3	0	0	0	0	11	3841	-48	-63	-56	0	0	-8	-3	
550	A42046	330	LIN Subassembly	3	0	28	0	28	2	216	0	0	0	0	0	0	0	
551	A42048	330	LIN Subassembly	3	0	0	0	0	25	2019	-31	-8	-9	0	0	-3	-14	
552	A42050	330	LIN Subassembly	3	0	0	0	0	75	9968	-655	-81	-291	0	0	-18	-9	
553	A42052	330	LIN Subassembly	3	0	6	0	6	144	2913	-91	-19	-30	0	0	-2	0	
554	A42054	330	LIN Subassembly	3	0	0	0	0	25	688	-26	-3	-10	0	0	0	-2	
555	A42058	330	LIN Subassembly	3	0	29	0	29	3	171	0	0	0	0	0	0	-1	
556	A42060	330	LIN Subassembly	3	0	0	0	0	-1	1455	-10	0	-1	0	0	-8	-1	
557	A42062	330	LIN Subassembly	3	0	0	0	0	0	1692	-7	-18	-28	0	0	-1	0	

Impact

Business Impacts

- Business impacts by automating/optimizing the planning process
    - Faster and more accurate decisions can be performed
    - Overall stock levels could be reduced by automating the planning process and improved efficiency
    - Less human planners would be involved in the process
    - Machining departments can rely on the machining plan and just follow the plan
    - Reduce backlogs
    - Reduced premium freight

Other Impacts

- Better work environment
- Improved morale of human planner
- Reduced Non value-added activities

Measures of Success

KPI´s to measure success of an automated planning tool

- Right part available in right quantity before each operation of bevel gear and pinion
- Optimized stock levels that retain the ability to absorb order variance
- Improved delivery performance to internal and external customers
- Timesaving in the planning process

The above KPIs can be measured by comparing results obtained via the automated planner with the human-derived plans using real or realistic customer order data.

Planning Integration

As the goal is to fully integrate the Planning tool in the planning process, we foresee that the exchange of data from the different input data sources should be once a day as defined during the project. The Planning tool will have a direct interface with MES (Manufacturing Execution System) and MES will communicate with ERP (Enterprise resource planning).

Glossary of Terms, Abbreviations and Acronyms

ERP	Enterprise resource planning
WMS	Warehouse management system
ASCP	Advance supply chain planning
EDI	Electronic data interchange
MES	Manufacturing Execution System
Takt	Declared time for one machining job
Make-to-Order	Produced in same sequence as customer demand or customer production line sequence
Make-to-Stock	Make-to-stock means that we aim to keep a fixed stock level between machining and assembly department.
EBS	E Business suite
APS	Advance planning system

MDS	Master delivery schedule

Requirements

Operations preparation

FM-010 - Optimal Plan Generation	**Type F/Q**	**Verif. T**
The UP should allow the generation of an optimal plan with regard to the inputted optimization function.		
Comment: The flexible manufacturing usecase requires that the jobs required by the EDIs (i.e. customer orders) are processed on the appropriate machines while respecting time constraints and satisfying optimization criteria. The optimization criteria is captured as part of the modeling, and can consists of minimizing machine wear, minimizing processing time, and miniming excess stock.		

FM-020 - Online replanning	**Type F/Q**	**Verif. T/A**
When simulating the execution of a plan, the TSB and UP shall provide functionalities to replan based on the current state of execution.		
Comment: When a plan is under execution by the operators, it is possible that an unexpected event occurs (machine breaks, new input). Operators must be able to trigger a manual replanning using the current state of the environment.		

FM-030 - Valid Plan Generation	**Type. F/V**	**Verif. T/A**
The UP shall consider inputted constraints (time & resources) when providing a plan.		
Comment: The FM use case requires the plan to consider a complex set of rules and resources (machines, time constraints). When a new order is provided, the operators should be able to input the updated set of constraints to see if a plan could be generated, and therefore the order can be accepted.		

FM-050 - Plan inspection	**Type. Q**	**Valid. T**
Given a plan and a problem definition, the UP should provide functions to inspect the plan and simulate the state of the problem after each action.		
Comment: The providers of the FM use-case requested that before following a plan provided by the UP, the operators can inspect said plan.		

FM-060 - Manual modification of a generated plan		**Type: Q**	**Verif. T/I**
	The operators should be able to modify a plan automatically generated by the UP		
	Comment: Because of the difficulty to model all existing constraints and expert knowledge, the operators should have the possibility to modify a generated plan in case some aspect of the process has been overlooked or missed.		

FM-070 - Parallel actions		**Type. F**	**Verif. T**
	The UP should support the use of concurrent actions that may create conflict in execution		
	Comment: For the sake of efficiency, actions that are performed using different resources are parallelized. Planning these parallel actions must respect the temporal constraints to avoid conflict / error during the process.		

T2.4 Logistic automation

Context

This section describes the logistics automation use-case, in particular related to the behavior of robots that are deployed by Magazino GmbH for logistics management in warehouses. In this scenario, robots are given a prioritized list of jobs with the goal of operating in a customer warehouse.

Jobs can be thought of as high-level imperative commands that are given by the Magazino coordination system to each one of the robots deployed on the field. Examples of such jobs are: pick up item X from compartment Y and store it in the onboard shelf, navigate to compartment Z, put down item X currently in the onboard shelf in compartment W, recharge on the charger C, etc. Such jobs are composed of one or more high-level actions that the robot will have to perform to complete a certain job. For example, in order to complete the job of "picking up item X from compartment Y and store it in the onboard shelf", the robot will have to navigate to compartment Y, search for item X, pick it up, and store it inside its onboard shelf. Instead, in order to complete the job of `putting down item X currently in the onboard shelf in compartment W`, the robot will have to navigate to compartment W, search for a free spot where to place Item X, pick up the item from the onboard shelf, and place it in the target position previously found.

The behavior of the robot performing jobs is currently defined encoded by hand-written policies using a behavior tree formalism extended by Magazino[3]. A single behavior tree, comprising a set of hierarchically composed subtrees, is responsible for the whole robot behavior. The behavior of the robot can be thought of as a loop, where the robot keeps checking a queue of jobs assigned to it and executes them one at a time. These jobs have a type, and each one of such types is associated with a hand-written plan, encoded using a behavior tree formalism. These plans are formed by robot specific subtrees, which can be thought as primitive action blocks with their own preconditions and postconditions, which are also represented as behavior trees.

Implementing the control of robots using behavior trees requires expert knowledge, is labor-intensive and thus exhibits a high probability of logical errors, programming errors and/or errors due to unforeseen combinations of environmental factors.

[3] https://patents.google.com/patent/WO2017148830A1/en?oq=WO+2017%2f148830

Planning Application

In the context of this project, the use of planning techniques is suitable to lighten the job of the behavior designer, by providing means to better analyze in advance the behavior subtrees, their conditions and execution, as well as to dynamically select the subtree(s) to be executed given a goal (job) and the situation. This is particularly useful in the presence of parallel actions (e.g., the movements of the robot base and of the gripper) and of failures / need for recovery actions.

The behavior trees currently used in Magazino robots are able to handle the following specific problems, some of which are typical in most autonomous robot scenarios:

- **Action failures and recoveries**: actions may have unpredictable effects, in particular they may fail to achieve their desired outcomes, and the reasons for the failures can be multiple. The system includes a set of recovery actions to be performed, depending on the situation and the reason of the failure.
- **Durative and parallel actions**: many actions, in particular those related to the movements of the robot, are not instantaneous and, in order to achieve better performance, our robots can parallelize some actions, in particular if they are related to different parts of the robot (e.g., the mobile base and the manipulator), or if it is possible to perform computation together with robot movements.
- **Unpredictable events**: during the execution of durative actions, some unpredictable event can cause the current plan to be invalidated, thus requiring the system to handle the anomalous situation before restoring the normal execution. In general, (a subset of) the state can change due to exogenous events, possibly requiring to slightly change the predefined nominal behavior or to completely abort the current execution and follow a new behavior.
- **Any-time restart capability and partially known state**: the robots are able to restore their functionalities after a power-cycle, this means that the state can be partially unknown at some point and the system requires some sensing action to collect the missing data.

Given the above requirements, we expect the automated planning technologies to be useful to the Magazino system in at least two different processes, as described in the following.

Offline design aid tool to check the soundness of the behavior definition

In order for automated planning techniques to be used in the current system, the action-related sub-trees should be identified and represented by the means of a planning formalism (e.g., PDDL). These action representations augment the current sub-trees defining the action execution. Since actions pre- and post-conditions are thus made explicit, an automatic mechanism can be developed to check the soundness of the sub-trees. Such soundness can be used to formally check if all the plan outcomes have been taken into account and to verify the correctness of every case scenario.

For this particular use-case, the planning technologies should consider at least the fact that actions may fail to achieve their nominal effects and that some actions are not instantaneous and may be performed in parallel.

Runtime reactive planner to compose the action subtrees

Once these sub-trees are augmented by a description in a planning formalism, they can be automatically composed at run-time using automated planner techniques in order to dynamically condition the robot behavior tree and choose the right subtree to use in every new situation. In other words, the planner can be configured to dynamically change the execution of the behavior tree by altering the composition of sub-trees and/or single nodes.

The planner in this scenario should be reactive to the changing environment conditions, to be able to repair the plan or even replan in response to unforeseen events and failures.

Current state («contingent data») is defined by:

- The current robot state, e.g., if it has an error, if it is in a functioning state, etc.
- Environment state, e.g., blocked corridors, (expected) position of items in shelves, etc.

- Job state, e.g., if the item of the job has been located, if it was picked or not, etc.

The planning technologies, in this case, can be particularly useful to handle unexpected events, the any-time restart capability, and the actions that are needed to reconstruct the state.

Background Section

Many planning approaches have been proposed over the years to deal with different assumptions concerning the environment and the agent. This section will give some pointers to relevant planning approaches that could be considered with regard to the requirements specific to this use-case.

First, the use case being based on behavior trees, it can be useful to review some work that considers the integration of planning and behavior trees. Planning and behavior trees have recently been considered together, for instance to automatically create a BT [1].

Due to their similarities with BT in the hierarchical representation of behavior, Hierarchical Task Networks also seem a relevant planning approach for the current use case [2]. Even though they don't yet address all the specified requirements, integration of HTNs and other planning approaches could be considered.

Non-determinism, i.e., the fact that the same action taken in the same state can lead to different results, has been widely studied in the literature, and many models and frameworks have been proposed over the years. For instance, non-deterministic planners for common domain languages such as PDDL, Linear Temporal Logic (LTL) [3], or contingent planning [4, 5]. All these approaches produce conditional plans that allow to account for the possible outcomes of an action.
It is important to note that other approaches prefer representing the solutions to a non-deterministic planning problem as policies instead of plans. While a plan represents a succession of actions, a policy represents a mapping from states to actions, given for each state the action to be performed. This is the case for Markov Decision Processes, in which the possible successor states of a given action are represented as a probability distribution, or FOND (Fully-observable Non-Deterministic) planning approaches, in which the possible successor states are represented as a set, for instance in [6]. In the case of a probabilistic distribution of action's results, non-determinism is also sometimes referred to as "stochasticity".

In many planning domains, action duration is not considered: either actions are assumed instantaneous, or the duration of the action does not matter for the planning problems. Durative actions, on the other hand, are used when action durations affect the planning problem, often because different durations have different effects on the environment or because some pre-conditions must be met during the whole action duration. Planning with Durative Actions have been incorporated to a limited extent in the PDDL planning language [7], and in different planners (for instance [8]). It has also been considered in stochastic domains [9, 10], and in the context of online planning [11].

Partial Observability in planning means that the deciding agent does not have full knowledge of the environment at all times of the plan execution. This can be due to limited sensing capabilities or missing data. The difficulty of partial observability lies in the fact that the search space is not a space of state anymore, but a set of "belief states" or "epistemic states", i.e., considering what the deciding agent believes or knows about the set of states. It has been included in specific planning approaches like in contingent planning, and-or search [12], or HTN [13].

In addition, one of the most common frameworks to deal with both is Partially Observable Markov Decision Processes (POMDPs). Another relevant framework is Epistemic Planning [14]. It is worth noting that many planning approaches under partial observability are still "limited" to reaching a desired goal state, but not explicitly reasoning about the planning agent's state of knowledge. On the contrary, Epistemic Planning usually deals with the problem of deciding which "epistemic action" to perform to go from one state of knowledge to another desired one. Epistemic

Planning is a desired behavior in this specific use case, as the anytime-restart requirement expects that the robot can "rebuild" an appropriate state of knowledge to carry on the plan.

References

[1] M. Colledanchise, D. Almeida and P. Ögren, "Towards Blended Reactive Planning and Acting using Behavior Trees," 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8839-8845, doi: 10.1109/ICRA.2019.8794128.

[2] Georgievski, Ilche, and Marco Aiello. "HTN planning: Overview, comparison, and beyond." Artificial Intelligence 222 (2015): 124-156.

[3] Camacho, A., Triantafillou, E., Muise, C., Baier, J., & McIlraith, S. (2017). Non-Deterministic Planning with Temporally Extended Goals: LTL over Finite and Infinite Traces. Proceedings of the AAAI Conference on Artificial Intelligence, 31(1).

[4] Hoffmann, J. and R. Brafman. “Contingent Planning via Heuristic Forward Search with Implicit Belief States.” ICAPS (2005),

[5] To Son Thanh et al. “Contingent Planning as AND/OR Forward Search with Disjunctive Representation.” ICAPS (2011)

[6] Geffner, T., & Geffner, H. (2018). Compact Policies for Fully Observable Non-Deterministic Planning as SAT. Proceedings of the International Conference on Automated Planning and Scheduling, 28(1).

[7] Fox, M. & Long, D. (2003) PDDL2.1: An extension to PDDL for expressing temporal planning domains. Journal of Artificial Intelligence Research

[8] Do, M. and S. Kambhampati. “Sapa: A Scalable Multi-Objective Heuristic Metric Temporal Planner.” (2002).

[9] Mausam and Daniel S. Weld. “Planning with Durative Actions in Stochastic Domains.” J. Artif. Intell. Res. 31 (2008): 33-82

[10] Weld, Daniel S.. “Stochastic planning with concurrent, durative actions.” (2007).

[11] Cserna, Bence et al. “Planning Time to Think: Metareasoning for On-Line Planning with Durative Actions.” ICAPS (2017).

[12] Bertoli, P. et al. “Strong planning under partial observability.” Artif. Intell. 170 (2006): 337-384.

[13] M. Weser, D. Off and J. Zhang, "HTN robot planning in partially observable dynamic environments," 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 1505-1510, doi: 10.1109/ROBOT.2010.5509770.

[14] Aucher, Guillaume, and Thomas Bolander. "Undecidability in epistemic planning." Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. 2013.

Introductory Example

As stated above, jobs, such as picking an item from a shelf or going to charge, are associated with specifically handcrafted subtrees represented as behavior tree plans. These plans are formed by robot-specific subtrees, which can be thought of as primitive action blocks with their own preconditions and post-conditions. Additionally, each of such plans needs to be crafted taking into account the possibility that the behavior tree executor might get shut down at any point in time. Such a requirement implies that the plan state information is lost and must be reconstructed at restart time. An example of such a type of plan is shown in Figure 1.

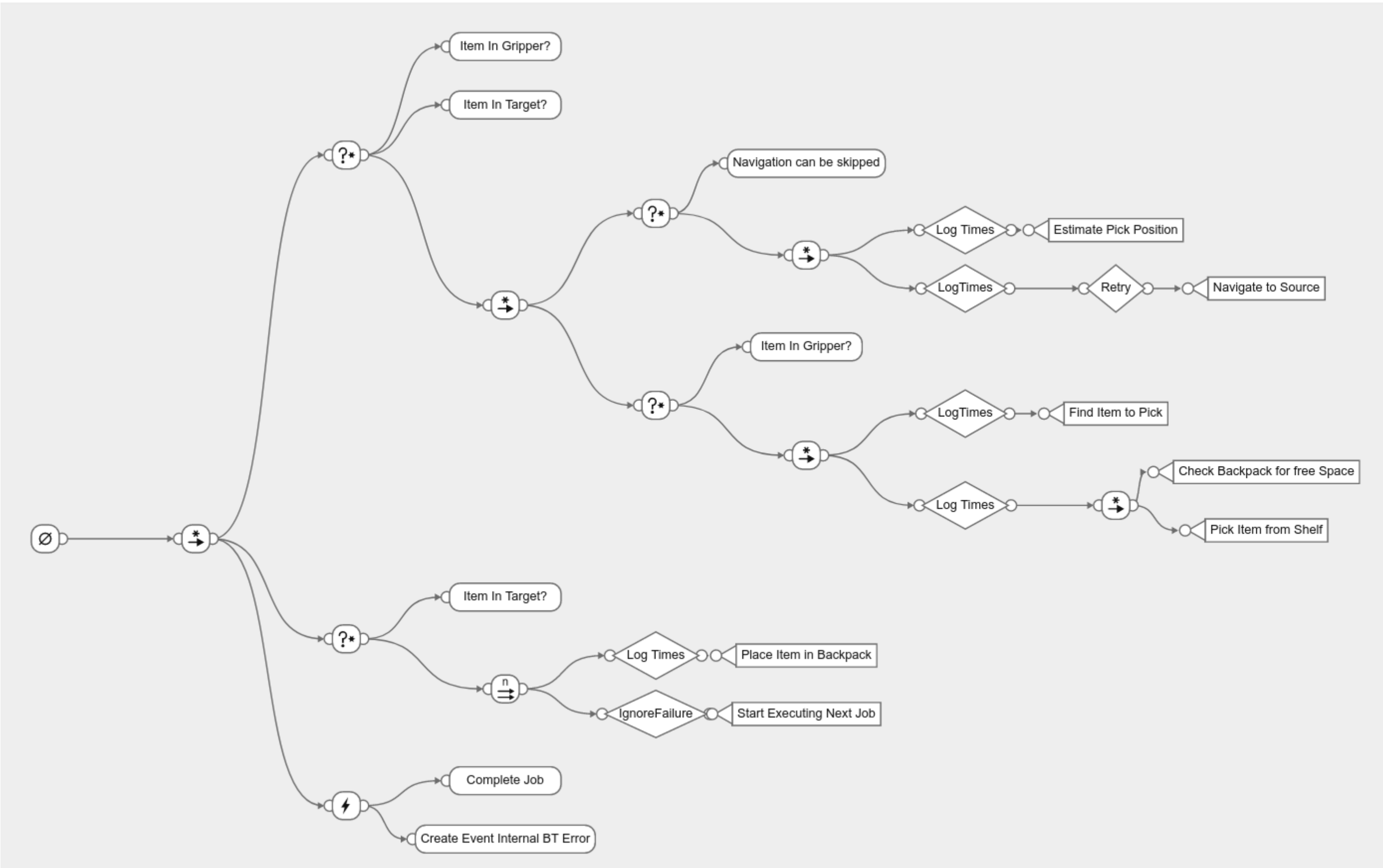


Figure 1: one of the behavior trees currently used in Magazino robots

In the image, the plan for picking an item is shown. The primitive action blocks are represented by subtrees like «Navigate to Source», «Find Item to Pick», or «Place Item in Backpack». Preconditions are implicitly represented by the sequencing of the subtrees (e.g., I cannot start picking an item from the shelf if I haven't successfully navigated there and found the item to be picked). Post-conditions are instead represented as checks that allow to 'skip' certain parts of the plan in certain conditions (e.g., if the item is already in the gripper, we do not need to navigate to the source shelf, nor find it, but we can directly proceed with putting it into the backpack).

In the context of AIPlan4EU, there are different approaches that can be followed to introduce the planning technologies in the system. In particular, the requirements and the constraints of specific jobs and the relevant actions can be encoded using some formalism consistent with planning techniques.

In this way, we can achieve two different goals:

1. During the design phase of the behavior trees, the user can be guided with a tool that is able to check if the tree satisfies some conditions, if some effects (e.g., some possible failures) has not been taken into account, if there are already actions that allow to handle the failures and recover the normal behavior, etc. (see Section 2.1).

2. During the execution of the plan, i.e., at run-time, another planning technology can handle exceptional events (i.e., the recovery from a shutdown) by considering the reconstruction of the knowledge of the world and the recovery from those unexpected events (see Section 2.2).

In principle, it would be also possible to combine the two goals and eventually make use of a single planner that is able to arbitrate among the different actions, to consider parallel executions, unexpected events and state recovery. However, the impact of this latter all-in-one solution should be carefully analyzed and considered only as a further goal, taking into account the possible consequences on the ease of designing and debugging behaviors designed with a hybrid solution like the one above.

Business impact and measures of success

The manual design of behavior trees is a long and error-prone task requiring experienced developers, continuous maintenance and trial-and-error phases. We expect the introduction of automated planning mechanisms to reduce and simplify this task and its cost. We consider two possible metrics to assess the success of the integration:

- we expect the complexity of the behavior trees to be reduced, in fact, the introduction of planning technologies should allow us to remove some condition checks and the decisions from the behavior trees because they are handled at a different level of abstraction; the complexity of the current behavior representation can be measured by the size of the behavior trees, e.g. by their total number of nodes: decreasing this number also results in an improvement in maintainability of the overall behavior specification;
- increasing the maintainability of the behavior specification should result also in a reduced amount of bug reports, related, in particular, to unforeseen conflicts between parallel actions and unforeseen failures: we could find a way to track down these specific bug reports and check if there is a reduction in the number or in the time spent to fix them.

In relation in particular to the offline design aid tool, we can consider, as a measure of success, the number of unforeseen situations that are detected (and that, without the technology, would have required a long process of manual running, detecting, debugging and fixing).

Planning Integration

Magazino behavior trees are composed in a hierarchical structure where subtrees are referenced in other subtrees as special nodes. This automatically induces a hierarchical structure of the trees that can also be used as a starting point for including the two use cases described in Sections 2.2 and 2.3. In particular, we envision to use the off-line tools to guide the construction of single-responsibility or simpler sub-trees (actions), and to compose their execution at run-time using the on-line planning technologies, as we showed in the picture in Section 3.
We foresee some steps that are required to integrate the outcomes of AIPlan4EU in our system, as we describe in the following.

Identify the behavior trees that are responsible for single functionalities. These are the trees that have been called “single-functionality trees” or “action trees” in the rest of this document.

Formalize the input data needed for the offline design aid tool. The design and maintenance of the single-functionality trees remains a task to be performed manually, but we want to include here the help of the offline tool described in Section 2.2: in order to accomplish this goal, we will identify (either manually or finding an automatic procedure) the overall tree goal, the pre-conditions, execution conditions and possible recovery mechanisms.

Integrate the offline design aid tool. Given the inputs identified above, the offline tool described in Section 2.2 will be integrated in our behavior tree editor as a check that can be issued by the designer to understand if the tree can be considered complete and correct, by analyzing its structure, possible executions and conditions. This integration can be realized by calling an API that, given as input the behavior tree structure and the data identified in the previous step, returns a set of problems and their possible solutions, that can be visualized directly in the editor. We expect the API provider to be executed next to (e.g. on the same server) the behavior tree editor.

Identify the decision variables that compose the "state" for the runtime reactive planner. At a higher level of our behavior trees, the single-functionality trees need to be composed online by the use of a planner and a set of decision variables that define the current state: the first step in this case is to identify such variables in order to present them to the planner.

Refactor the behavior trees in order to include the runtime planner. We plan to replace some of the trees that we use to compose the single-functionality behaviors with a planner, as it is shown in Figure 2.

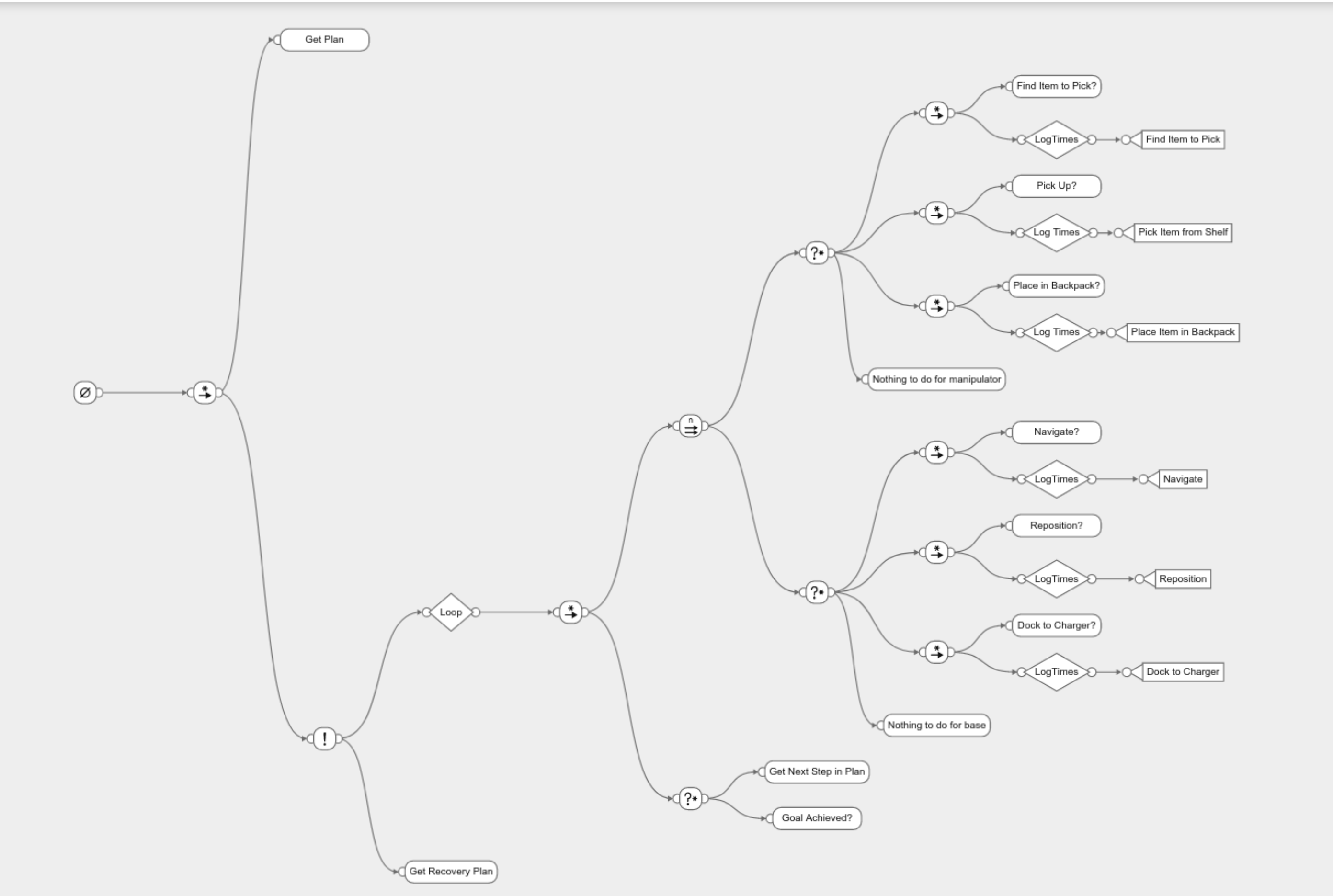


Figure 2: how we expect to refactor some high-level behavior trees, in order to include the planner(s)

In this reorganization, the planner is included as a node in the tree (called "Get Plan" in Figure 2) that gives as output the action or set of actions to be active in any given step of execution. The behavior tree then executes the subtree(s) related to these actions (the rectangular nodes on the right side of Figure 2).
The possible unexpected deviations from the nominal execution of the plan (e.g. unexpected events) can be taken into account in the "Get Plan" itself, thus representing a planner that is able to deal with all the specific features listed in Section 2, for example repairing the plan, requesting for recoveries, etc., or these "special situations" can be taken into account in another planner (represented as the "Get Recovery Plan" at the bottom of Figure 2) if we consider that it is better to split the problem. We expect, in this case, that the planner can be executed on the same machine where the behavior tree executor is running (e.g. on a robot).

Example and Evaluation Data

Data for the interaction with the planning technologies

As we described in Section 5, we provide for two interfaces between the current system and the planning technologies. The first one is related to the offline design aid tool, and we envision an integration in our current behavior tree editor that can be triggered by the user to check the correctness of the behavior tree being. We can provide the structure of the tree (that currently is defined using the JSON format) and the relevant state variables and conditions to be checked.

For the online reactive planner, on the other hand, we need to define a formalism that is able to specify the domain, the current state of the system and the current goal with the ability to include the problem-specific traits as described in Section 2. The expected response to a request defined in this way is the current (parametrized) action or set of actions to be started, to be kept running or, possibly, to be interrupted/aborted.

Data for the evaluation of the resulting system

As described in Section 5, in order to measure the impact on the system and its maintainability, we should be able to provide:

- the current number of behavior trees and the total number of nodes;
- a measure such as the current bugs / the time spent on fixing bugs that can be automatically detected or completely avoided by the use of the technologies described in this document: we are still looking for some facilities to easily identify this kind of bugs.

Requirements

Offline Design Aid Tool for Behavior Trees

LD-010 – Automatic generation of an execution (plan) that violates some constraints	Type: **F, V**	Verif: **T**
The AIPlan4EU Framework shall allow the automatic generation of a sequence of executions in the behavior trees that violates some constraints		
Comment: with this feature, at design time, we are able to detect possible execution of the trees that leads to errors		

LD-020 – It is possible to generate (in)validation plans in a feasible time	Type: **PE**	Verif: **T**
The time required to analyze a behavior tree searching for an execution that violates some constraints should be reasonable		
Comment: ideally, the analysis of a single tree should require less than a minute, or, as an alternative, the analysis of the whole set of behavior trees should require less than a hour (to be run nightly, for example)		

LD-030 – The behavior tree design process should be less error-prone: less bugs in production environments	Type: **BI, M**	Verif: **A**
In order to be useful for the end user, the automatic analyzer should lead to better designed trees		
Comment: potential bugs are discovered at the design phase instead of when the behavior trees is running in a production environment		

Logistics Automation: Runtime Reactive Planner

LR-010 – Automatic plan generation	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the automatic generation of plans that provides a sequence (or partially ordered sequences) to reach a goal		
Comment: the on-line planner should be responsible to orchestrate among high level actions		

LR-020 – Parallel actions	Type: **F**	Verif: **T**
The generated plan should be able to deal with the concurrent execution of potentially conflicting actions.		

	Comment: in order to increase the performance of the system, the actions that are not related to the same resource (e.g. the robot base or the manipulator) should be parallelized

LR-030 – Goal priority and conditions	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the specification of goals with different priorities		
Comment: a goal should be taken into account only when the higher priority ones have been satisfied, in our domain these goals also can be enabled/disabled given some conditions		

LR-040 – Optimal plans	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow the generation of optimal plans		
Comment: optimal plans are important in those states where multiple actions are possible		

LR-050 – Online planning	Type: **F**	Verif: **A**
The plan needs to be generated, followed, repaired in online operations		
Comment: the planning/execution loop should be fast enough to be followed in online operations		

LR-060 – Repairing plans	Type: **F**	Verif: **T**
The AIPlan4EU Framework shall allow to repair plans that have been generated before in case some conditions make them invalid		
Comment: in case of unexpected and undesirable situations, the planner should be able to repair the current plan being executed		

LR-070 – Reduction of complexity and maintainability	Type: **BI, M**	Verif: **A**
Thanks to the introduction of automated planning technologies, the size of the behavior trees decreases, thus resulting in less complexity maintenance efforts		
Comment: some of the conditions that are currently explicitly checked in the tree itself can be moved to the domain definition, thus dumping their check on the planner		

T2.5 Shuttle fleet management

Context

EasyMile is a company that focuses on autonomous driving and develops solutions for tow tractors operating on airport tarmacs and factories for logistics transportation.
In this domain, planning is identified as the technology of choice to coordinate fleets of vehicles to ensure a continuous operation mixing planned and on-demand requests. In factories, a logistics flow needs to be available 7 days a week, 24 hours a day, while minimizing the number of vehicles needed.

Moreover, using electrical AV implies:

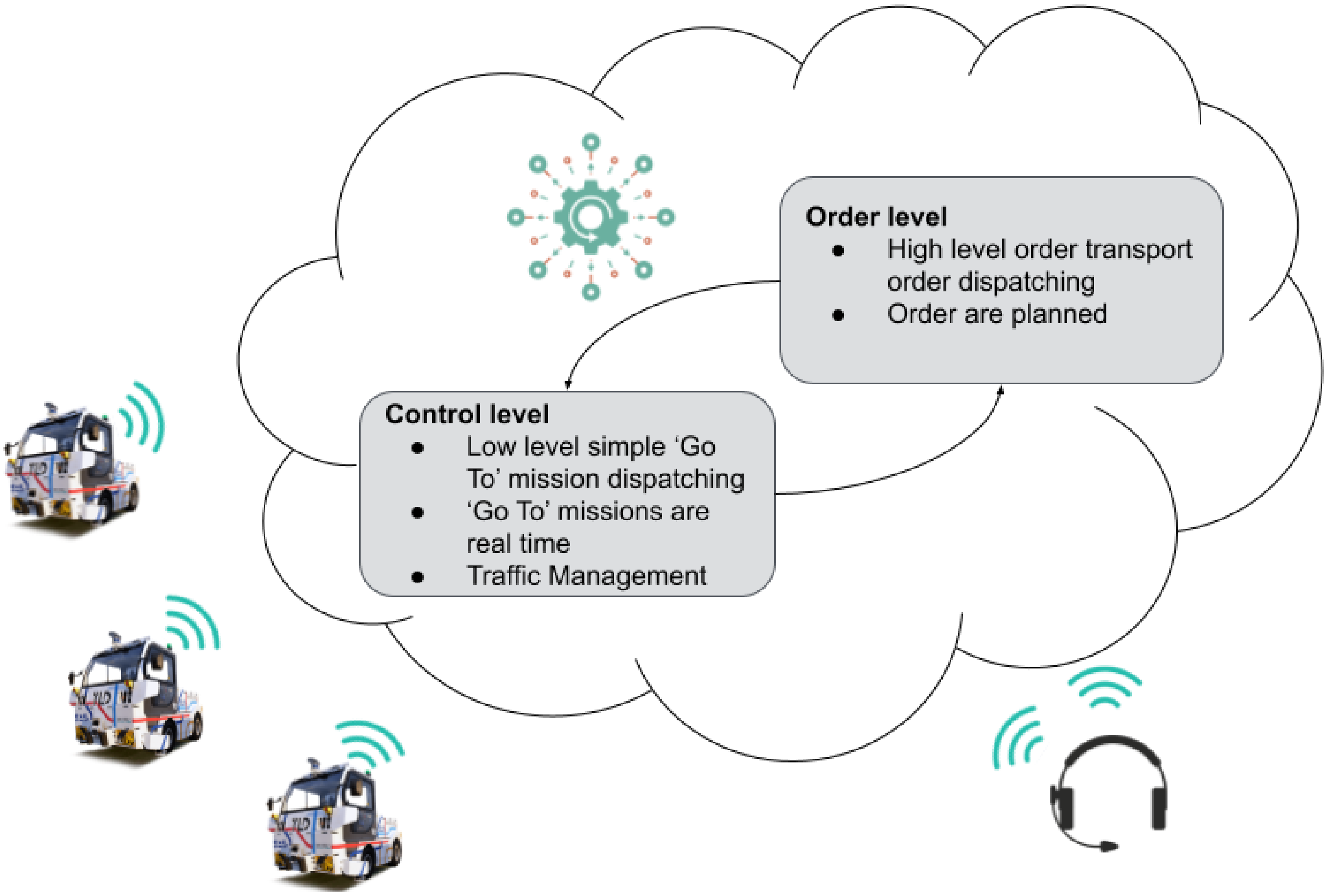
1. The vehicle must charge when they are not towing loads, i.e. in between missions
2. Not to charge all the vehicles at the same time, to guarantee a minimal service rate
3. Optimizing the use of the charging points that may be expensive per unit

A TractEasy, EasyMile's autonomous tow tractor (source https://easymile.com/)

EasyMile's Fleet Management System is able to manage a fleet of TractEasy vehicles in order to make them realize payload transportation tasks. On an operation site are defined several points of interest:

- Stations, at which the vehicle carriages will be sent to either be loaded or unloaded
- Charging points, at which the vehicle will be charged

The following figure provides an overview of EasyMile's Fleet Management System capabilities in the context of logistics transportation.



The "Order level" manages the high level objectives of the fleet, plannable metier-oriented tasks. It communicates with the "Control level", that manages simple motion missions (such as going to coordinates), that are vehicle specific and must be defined in real time. The Control level also implements an automatic traffic management system that is out of this use case scope.

Advanced planning capabilities are needed at the "Order level". In a logistics context, a transport order is a list of actions a vehicle must realize. Actions start by going to a pick-up station to be loaded, and end by going to a drop off station to be unloaded. In between the first pick-up and last drop-off actions, a finite number of pick-up/drop-off actions can be defined.
An order can be defined for a given vehicle, or not, in which case the Fleet Management System must select the best vehicle to realize the order.
Orders are planned using a queue, in which the rank determines the order priorities.

The general operational workflow is the following:

1. A remote operator defines a set of orders
2. The order level assigns the orders with the highest priorities to the available vehicles:
    a. It defines simple 'Go To' missions at the correct time
    b. It monitors the missions executions, eventually creates new ones to continue the order
3. When some order executions are completed, the Order level realizes the next waiting order with the highest priorities
4. The Order level decides when to charge the electric vehicles so that a minimal execution flow is maintained:
    a. Charging orders are automatically created and added to the waiting order queue
5. The remote operators can create new orders and edit the waiting order priorities at any time

A charging order is vehicle-specific and ends in a simple 'Go To' mission, for which, once the destination is reached, the vehicle must enable its charging component. TractEasy vehicles are electrically powered. The charging technology is using induction. The vehicle has to move to precise coordinates in order to be able to charge wirelessly. Due to some physical constraints, any charging point cannot charge all vehicles. For that reason, as a fleet configuration, each vehicle is associated with a given charging point.

As a final complement, a transport order can be defined with the label 'repeating', in which case, at execution, the order is duplicated and the duplicate is added to the end of the order queue. As the duplicate has the same property, it will also be repeated as well. This allows the user to have some kind of perpetual order definition.

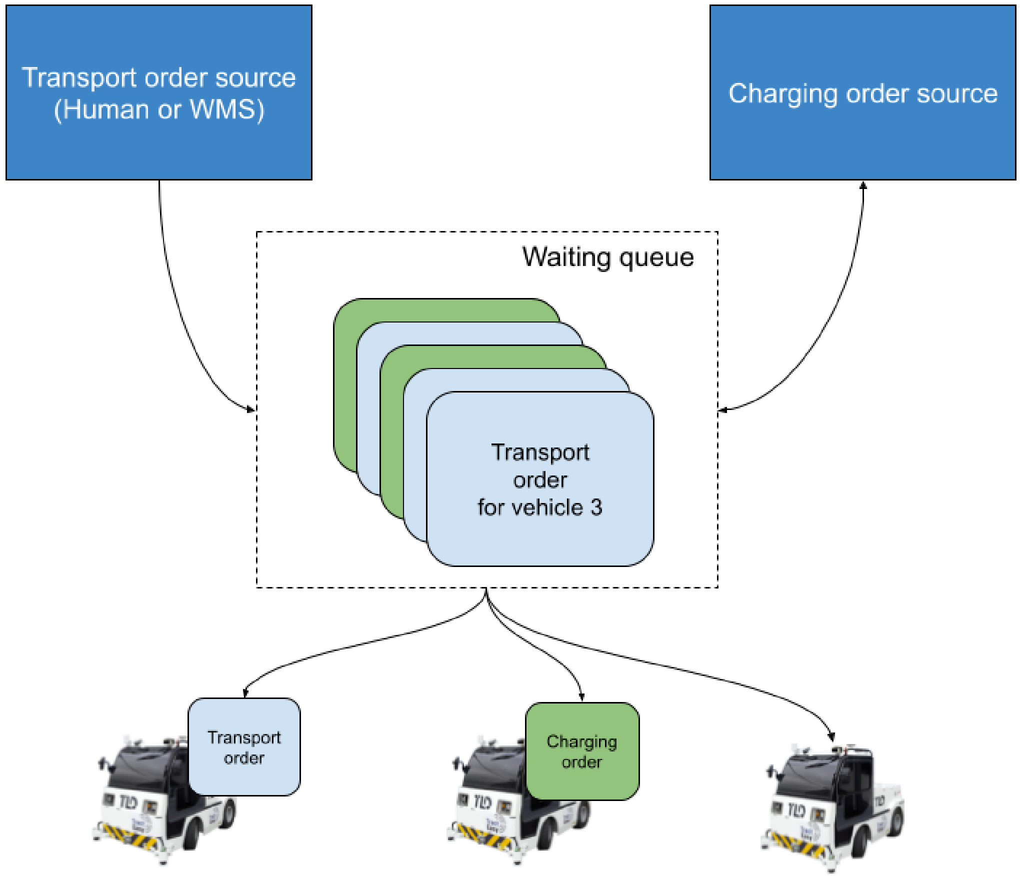
Planning Application

This use case is focused on using advanced planning capabilities to improve the general workflow step 4 and 4.a. Today's solution is pretty naive: once a vehicle battery level falls below a certain threshold, a charging order for that vehicle is created with the highest priority possible (higher than any other transport orders). As a consequence, once a vehicle completes a transport order, its next order to realize is its charging order.
The operator can always change the charging order priorities and manually create charging orders to refine the planning, but the goal is to reduce the operator mind overloads and to let them focus on transport order planning.

Today's solution is too naïve in regard to the fleet charging time distribution. Starting the operation with all vehicles fully charged will end with all vehicles charging (or waiting for an available charging point) at the same time, while no transport order will be realized, interrupting or gravely reducing the flow.

In a nutshell:

- Transport orders are defined by a third party (human or unknown sources) and added to the waiting order queue. A transport order can be designed for one specific vehicle or any vehicle. A transport order can be defined as 'repeating' to enable perpetual missions.
- Charging orders are defined automatically using planning techniques and added to the waiting order queue. A charging order is designed for one specific vehicle.
- Ranks in the waiting order queues define order priorities.
- When a vehicle can, it realizes the order with the highest priority in the waiting order queue.



Order assignment

An order assignment algorithm eventually assigns an order to a vehicle when the vehicle becomes available. When a vehicle is available, the algorithm simply assigns the first order it found that is eligible to the vehicle.

As an example, let imagine a situation with two vehicles (vehicle 1 and vehicle 2), and the following waiting queue:

Priority	Order ID	Order type	Order eligibility	Repeating
Highest	79	Transport	Vehicle 1	No
	25	Transport	Any	No
	29	Charging	Vehicle 1	N/A
Lowest	42	Transport	Vehicle 2	No

Scenario 1

Vehicle 1 becomes available first -> the transport order 79 is assigned to Vehicle 1.

Then the following can happen:
1. Vehicle 2 becomes available -> the transport order 25 is assigned to Vehicle 2, or
2. Vehicle 1 becomes available again (before Vehicle 2) -> the transport order 25 is assigned to Vehicle 1

Scenario 2

Vehicle 2 becomes available first -> the transport order 25 is assigned to Vehicle 2 (as only Vehicle 1 can realize order 79).

Then the following can happen:
3. Vehicle 1 becomes available -> the transport order 79 is assigned to Vehicle 1, or
4. Vehicle 2 becomes available again (before Vehicle 1) -> the transport order 42 is assigned to Vehicle 2

This algorithm cannot be changed, the planning function focuses on insertion of charging order in the waiting queue (see next section).

Planning

The planning capabilities focus on creating charging orders for vehicles and choosing their priorities in regard to the transport orders, with the aim to maximize the number of transport order executions at any time.

Inputs

The planner has at any time access to:
- The waiting order queue
- The order currently executed by each vehicle
- The order full definition (routes, all stop coordinates and waiting times)
- The vehicle telemetry (including in between stop route progression, battery percentage, etc.)

Prediction

A battery prediction model could be computed. The ground can be considered as flat in warehouses, factories and airport tarmacs. The payload tracted by vehicles can be difficult to predict. If needed, another prediction model focused on the order duration could be computed.

Optimization

The typical situation that is wanted to be avoided is when N vehicles have not enough battery power to realize more transport orders, but there is M, with M < N charging points. In that case, N - M vehicles will have to wait for a charging point to be free, while not realizing any transport order.

What is wanted is to maximize due time.

A simple metric reflecting the payload flow is the number of vehicles executing a transport order at any time. Hence the planning should focus on maximizing that number over the time.

Decision

Planning should be able to provide inputs on when to create a charging order for a given vehicle, and on what rank in the vehicle order queue (reminder: ranks define the priority).

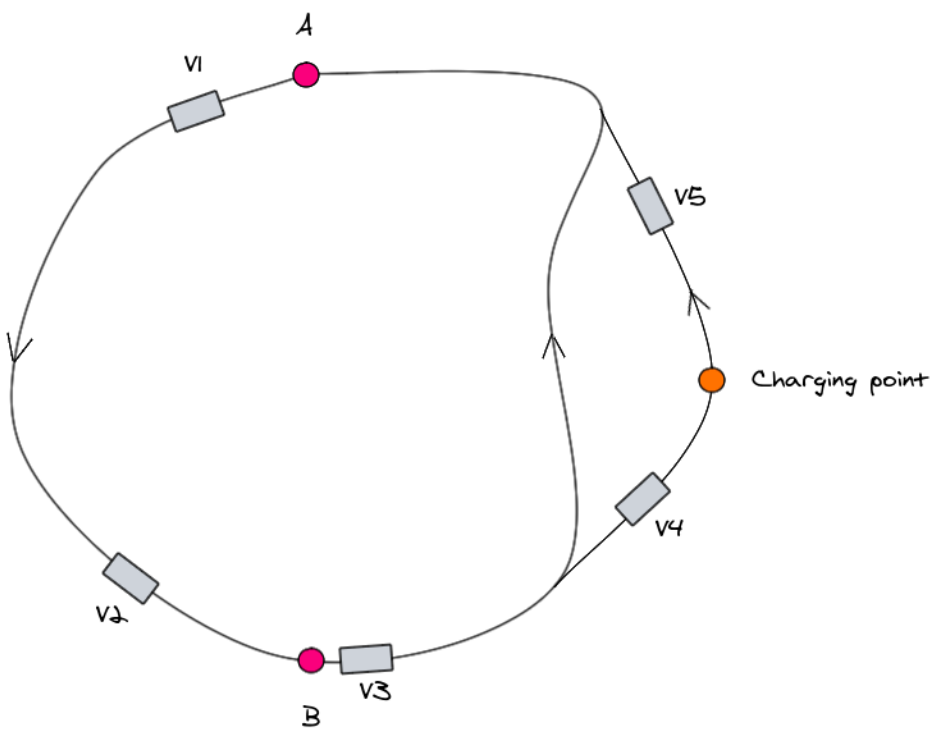
Using the waiting queue of the example above, planning could decide to add a charging order for vehicle 2 on the top of the queue (i.e. with the highest priority).

Frequency

The charging order source must trigger the planning function at least when a vehicle battery percentage decreases significantly (for example, 1 point) or when a new order is created or an order status changes (for example, from waiting to executing or from executing to finished).

Introductory Example

We consider a simple example where 5 vehicles need to continuously pick-up some payload at station A and drop-off the payload to another station B on a one-way road, while a single charging point is available for all vehicles.



Picking up and dropping-off payloads take approximately 1 minute and do not consume battery. Going from station A to station B (and B to A) takes approximately 20 minutes and 8% of a vehicle battery. We ignore traffic jams in that example.

The charging point can charge only one vehicle at a time. The charging equipment approximately charges the vehicle battery at the speed of 1% per minute.

Daily work starts at 2 AM and ends at 11 PM.

At the beginning of operations, the waiting order queue is defined as:

Waiting order queue					
Priority	**Order ID**	**Order type**	**Order eligibility**	**Actions**	**Repeating**
Highest	0	Transport	Any	[PickUp(A), DropOff(B)]	Yes
Lowest					

Then, all the vehicles are fully charged and placed at station A in a row, then set as “available” to realize orders. Order 0 will be assigned and executed by vehicle 1, and duplicated into order 1, which will be assigned and executed by vehicle 2, and duplicated into order 3, and so on until the five vehicles are executing an order and one transport remains in the queue: the fifth duplicate:

Waiting order queue					
Priority	**Order ID**	**Order type**	**Order eligibility**	**Actions**	**Repeating**
Highest	5	Transport	Any	[PickUp(A), DropOff(B)]	Yes
Lowest					

Executing orders					
Vehicle	**Order ID**	**Order type**	**Order eligibility**	**Actions**	**Repeating**
V1	0	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V2	1	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V3	2	Transport	Any	[PickUp(A), DropOff(B)]	Yes

V4	3	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V5	4	Transport	Any	[PickUp(A), DropOff(B)]	Yes

Once vehicle 1 will complete its order, it will be assigned to order 5 and so on.

With our current naïve approach, eventually, all the vehicle battery levels will be under 15% at almost the same time. That would provoke the automatic creation of a charging order for all the vehicles with the highest priorities, resulting in the following waiting queue:

Waiting order queue					
Priority	**Order ID**	**Order type**	**Order eligibility**	**Actions**	**Repeating**
Highest	47	Charging	V5	Charge(Charging point)	N/A
	46	Charging	V4	Charge(Charging point)	N/A
	45	Charging	V3	Charge(Charging point)	N/A
	44	Charging	V2	Charge(Charging point)	N/A
	43	Charging	V1	Charge(Charging point)	N/A
Lowest	42	Transport	Any	[PickUp(A), DropOff(B)]	Yes

Executing orders					
Vehicle	**Order ID**	**Order type**	**Order eligibility**	**Actions**	**Repeating**
V1	37	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V2	38	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V3	39	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V4	40	Transport	Any	[PickUp(A), DropOff(B)]	Yes
V5	41	Transport	Any	[PickUp(A), DropOff(B)]	Yes

Once vehicle 1 will complete order 37, it will be assigned to order 43 and go to charge itself. Once vehicle 2 will complete order 38, it will be assigned to order 44 and go to charge itself, but as the charging point is taken by vehicle 1, it will have to wait behind it. Other vehicles will end in the same state. So there is a lot of time where 4 of the 5 vehicles won't either charge or execute a transport order.

It is expected that the planning will provide input to create vehicle charging orders before the naïve approach would, so that vehicles would charge themselves sooner and no time where a vehicle is not charging or executing a transport order is observed.

Impact

Business Impacts

Logistics has already seen the rise of autonomy inside warehouses: Automated product line, Automated Guided Vehicles (tuggers, forklifts, unit loaders) or Autonomous Mobile Robots. The benefits of such automation are numerous: economical benefits (improving the uptime service time while decreasing costs), safety increase (elimination of dangerous manual operations), environmental benefits (optimal use of vehicles).

Transporting goods with a vehicle represents a significant part of the expenditure of a manufacturing company and is also the core business of third-party logistics companies. In the past years, we have seen a strong interest in the development of logistics services, but usually they are restricted to operating in indoor environments. Hence its applications for automation are limited to specific workflows requiring the use of a human driven vehicle to move in between buildings; an example would be to transport goods from inside a building and move the goods into another building on the logistics site.

The deployment of a fleet of autonomous vehicles as the TractEasy in logistics, can provide further benefits while not only increasing operational speed or handling more complex missions, but also exploring outdoors operations, line haul transportation, and last-mile delivery. Operations on industrial sites or airports, where security is already an important subject, can speed-up the deployment of specific on-demand services for logistics.

Beyond the autonomous vehicle itself, fleet management is crucial to ensure the successful integration and efficiency of autonomous connected vehicles in the logistics industry: real-time telematics allows fleet management software to control and optimize all associated tasks on the fly. From an economical point of view, the main goals are improving efficiency and productivity while reducing costs associated with inefficient operation (staff and operational costs).

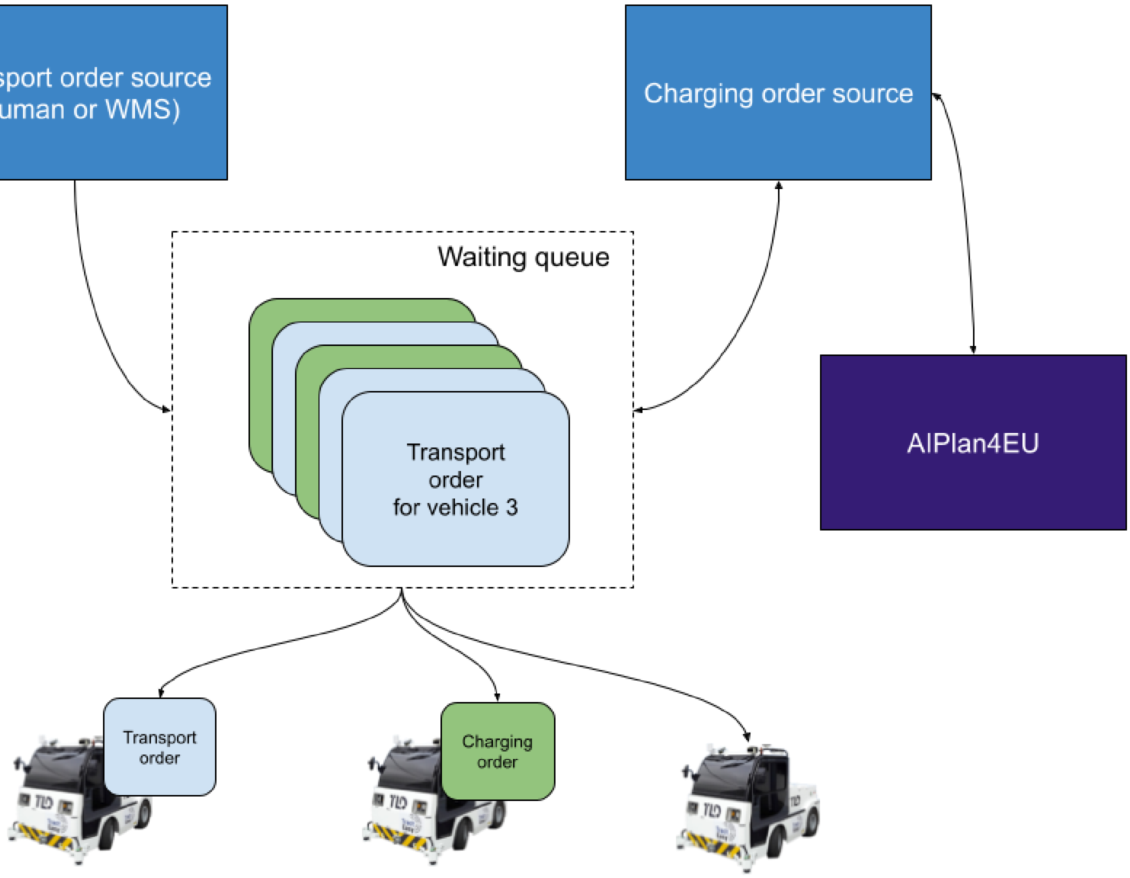
Planning will help us to keep a real-time mission control which ensures the optimal usage of resources in order to keep with the pace of operations. On the one hand, the space constraints as usage of the occupancy grid requires control of each AGV meter after meter with a fleet management system. Moving from AGVs to AVs requires deporting the mission management to the vehicle to allow it to keep operating even if connectivity fails. Planning needs also to focus on avoiding any potential deadlocks when assigning missions that may face disconnections instead of fine controlling grids. On the other hand, the execution of missions may encounter problems (obstacles, temporary localisation problems, vehicle failures). Planning could also help to handle such events to ensure a constant flow of vehicles in order to avoid interruptions of service.

Measures of Success

The two planning functions (naïve and AIPlan4EU) can be compared using the same test scenario inputs via a KPI indicating the number of transport orders realized in a given time period. AIPlan4EU integration is successful if more transport orders are realized than using the naïve algorithm.

Such KPI can be computed by EasyMile data service products for example, or by a neutral third party using EasyMile logistics orders HTTP JsonAPI API.

Planning Integration



In regard to the architecture described in section 1 and 2, it is planned to integrate the planning optimizer to the charging order source, a module that aims at automatically creating vehicle charging order with the maximal priority.

Today that module implements the naïve behavior: once a vehicle battery level falls below a certain threshold, a charging order for that vehicle is created. Tomorrow it will use the AIPlan4EU planner to know when to create charging orders.

Assuming the AIPlan4EU planner integration will be done with a static library call or a web server call, a call can be done any time the waiting order queue is updated.
The call arguments are the ones defined as inputs in section 2, i.e.:

- The waiting order queue
- The order currently executed by each vehicle
- The order full definition (routes, all stop coordinates and waiting times)
- The vehicle telemetry (including in between stop route progression, battery percentage, etc.)

In addition, a battery charging/discharging model must be learned:

- If the AIPlan4EU planner is stateless, that model must be constructed in the “Charging order source” module and given as input.

- If the AIPlan4EU planner is stateful, it should construct that model and reference it. Its reference should be given as input.

The AIPlan4EU planner output should be the list of vehicles for which the “Charging order source” module should create a charging order.

Requirements

EM left the consortium before the revision of this deliverable. As such it was not possible to formalize the requirements for this use-case.

T2.6 Automated experiment design

This document describes the proposed application of the Universal Planning Framework (UPF) in the context of Collaborative Robots (Cobots) applied in Research & Development (R&D) labs for Fast Moving Consumer Goods (FMCG) products. In this scenario, we have a large number of testing procedures and quality controls applied to many different product families, very often performed manually by human operators. The usage of cobot can greatly reduce repetitive and manual operations, and significantly improve the quality and reliability of the output data. But robot programming and dealing with a complex dynamic environment is the key limitation to broad adoption. With our work in this AIPlan4EU case study, we will explore the application of UPF to some selected operations; our vision is to prove that UPF can have a significant measurable impact along the key performance vectors:

- productivity - the overall procedure is faster and more efficient vs a hard coded operation
- robustness - the system can autonomously deal with unexpected events and errors
- user friendliness - non-specialized operators and users can modify and expand the robotics operations

Context: Adoption of flexible robots in FMCG research labs

The objective of Research & Development (R&D) departments in FMCG is to grow the business through innovative products that delights consumers in a broad range of applications and scenarios. The area of product testing is broad and complex, involving thousands of procedures and rich sets of output data that are used to drive the development and choice of new products to launch. Many of these testing procedures aim at mimicking what our consumers would do at home, for example testing a surface cleaner product to remove tough soils from a kitchen top: the testing procedure replicates the mechanical scrubbing of the human arm joined with the cleaning action of the product chemistry, the final outcome being a technical measure of cleaning performance. Many of these procedures are carried out manually by people in our labs, a task that is time consuming and prone to human variability. The vision is to leverage more and more robotic units to carry out the testing procedures, in cooperation with human researchers. The usage of robotics can boost the repeatability and accuracy of results, thanks to their accurate trajectory and dynamic controls, as well as throughput, being able to operate 24/7 and allowing a reliable numbering up. However there are a few significant challenges in this process:

- **Flexibility** - since we are in a research environment our testing procedures change and evolve very often, coping with new products, new applications, new trends. It’s therefore quite difficult to define exact protocols that specialized robotics/AI/planning engineers can implement once and for-all. We have a need to be able to constantly modify and adapt them, ideally non-skilled lab operators should be able to make the intervention themselves without the need for advanced training
- **Time and Resources For Development** - we currently have thousands of methods, and very few specialized resources that can develop advanced robotics applications. We would like to come up with an intuitive development platform, where non-robotics/AI scientists with minimal training can assemble robotics operations by combining atomic capabilities (e.g. object detection through vision, picking objects, measurement activities) in high level testing procedures that can be deployed in a short time
- **Interaction with Human Operators** - in several situations we would need to put human tasks in the loop, so that the robot could collaborate with humans, with tasks that are shared or where the human can change the course of action. This makes “hard-coding” of testing procedures even more difficult, as the number of possible options and nodes in the procedure graph increases exponentially.

Planning Application

A predictive model of the system is available: the entire process and the specifications of the actions to be performed at each step are known and can be modeled with some planning formalism. However, during execution, some non-nominal conditions may happen that require the plan to be corrected or generated again.

In other words, we consider situations where the nominal plan, generated by a planner according to the domain model, will be successfully performed in nominal conditions (i.e., when during execution, the environment evolves as modeled). However, we also want to consider some non-nominal conditions that happen when something not modeled in the domain occurs. In this case, we need to define: 1) a mechanism for the system to recognize such non-nominal conditions, 2) a procedure to adjust/repair the plan and recover it to reach the goal with minimal deviation with respect to the nominal plan.

In this context, we will consider human operators in the loop. Indeed, human experts present in the environment during the execution of the task, can easily detect non-nominal conditions that will probably lead to a failure of the current plan. In these situations, we envision an effective human-AI interaction to allow the human operator to inform the AI Planning system about non-nominal conditions and possibly teach it (in a natural way) how to recover from such situations.

Introductory Examples

To clarify the type of problems we encounter in this case study, we provide below a list of various practical examples of robotics applications in research labs, together with a description of the key challenges and ideal end state. These cases have been selected to provide simple case studies that can be formalized under a more general framework to enable broader applications (and re-applications) of AI planning techniques.

Quality testing of unit dose pouches

Soluble unit doses (see image to the right) are a very convenient form for dosing liquid products, for example in laundry washing context. They are subject to a very strict set of safety and stability rules, so several testing methodologies are applied to make sure these products fully comply with the required specifications. Some of the most common measures that are typically run are:

- weight
- dimensions
- elasticity of the film
- tightness of the pouch
- strength of the pouch (resistance to compression)

Each of these characteristics is measured with a different instrument, and each instrument can have several different settings; for example, the measure of strength can happen in several different ways (constant compression force, constant compression rate, a mix of the two, etc.), so the overall testing procedure involves several steps that can change and evolve in each testing session. Today we have human operators that handle the overall procedure that is a manually intensive operation, employing several people for entire days. The typical operations that people carry out are:

- picking up of individual pouches from product containers
- identification of pouches (production codes, production line, etc.) and labeling in the data system
- decide the sequence of testing based on the pouch type and project
- placing each pouch in the specific instrument
- setup and operation of the instruments
- record results
- record visual observations depending on the test outcome, for example if the pouch broke under compression detect where the breakage happened, in which compartment, detect type of failure
- dispose the pouch

Currently, any change in the overall procedure needs to be hard coded in the robot program, and there is no live interaction between the robot and human operators present in the lab. We would like to leverage an advanced AI planning system that enables lab operators with minimal training to be able to modify the procedure and adapt it to the needs at hand. Another current limitation is that the overall procedure is not optimized, being a fixed predefined sequence of individual operations.

Therefore, there is a significant opportunity to improve productivity by applying planning at two different levels:

1. <u>high level optimization and planning</u> - testing hundreds of samples in different instruments can be very time consuming with a fixed procedure. Very often we do not need all the measures for all pouches, and different

measures (i.e. different instruments) have different time durations. We would like to apply AI planning to optimize the combination of operations. The bottle-neck is the robot arm operation, and we aim at optimizing its use also allowing different pouches to be tested in parallel in different instruments at the same time;

2. improvement of each atomic operation - each individual operation (e.g. pick a pouch and place it on the scale) has been hard-coded using the UR polyscope software, very often by manually creating a trajectory of the robot arm. This approach has ample margins for improvements with AI planning. On top, having a planner for individual motions in our complex space (robot arm and several different instruments and/or even human operators) would allow us to adapt the approach to new lab configurations without an expensive manual re-programming effort.

The current setup, i.e. a robot arm with 6 DOFs and a fixed base, could be expanded by the addition of a mobile base. This would enable an even larger degree of automation, for example the robot unit could navigate our labs and collect samples to be tested autonomously, and interact with an even larger number of instruments.
In the context of the current project, we will start with a simplified approach, i.e. the static robot arm and a simplified set of planning choices (e.g. sequential operation, few desired/undesired events) and gradually increase the complexity to the full lab configuration and possibly even the mobile base. We will also study an extension using multiple robot arms.

Hand dish and surfaces washing performance test

In the field of Home Care there are several manual performance tests that are very difficult to automate; we need to mimic the interaction of human consumers with products (e.g. the dish detergent) a support device (e.g. a sponge) and the class of items that will be treated with the application (e.g. the specific dish to be cleaned). Humans have the capability of a highly dexterous manipulation, coupled with a multi-sensorial feedback; in the case of hand dish washing vision is used to detect areas with tough soil, as well as detecting shine and gloss to identify where further cleaning is needed, and finally hearing and skin feel both sends signals to the brain to connote real cleaning (e.g. “squishy-clean” sensation) as opposed to conditions where an invisible oily layer remains. For these reasons it’s quite complex to develop robotics testing procedures that can correctly reproduce the efficacy of a cleaning product as a human would do it.
This capability is fundamental for the overall product development effort; we need to be able to discriminate between subtle formula variations in order to find and optimize new products to be launched on the market.

The main difficulty we encounter in this application area is the complex robotic control problems; in general, we are faced with the task of controlling the application of a product and/or cleaning action (brushing, rubbing, etc.) mimicking how a human would do it. This typically involves force control and a multi-sensorial feedback, mainly image and sound data. Typically we don't have very good dynamic models, as the rheology of cleaning products, foam, cleaning implements (brushes, sponges, etc) makes it very difficult to generate one. And then the application surface can also be non regular, think about the shape of a typical plate or drinking glass, so a traditional model based force control is not trivial. The vision here is to have a planning engine and a learning model that could autonomously learn how to generate the cleaning trajectories and adapt them as the item to be cleaned changes.

Handling of raw materials inventory in a formulation lab

We have several formulation labs, where chemists and lab operators assemble a number of raw materials to create product prototypes, for example a laundry detergent that is created from a mix/blend of several surfactants, active ingredients, colorants and perfumes in a water base. This procedure is repeated thousands of times each day, to create unique formulations with a variable number of components (10-100 raw materials in general). Some parts of this process are automated, but we still have a significant portion that is handled manually, and one of the

time-consuming activities is the management of raw materials inventory and fetching of required raw materials to complete the formulation work. These are typically stored in bottles and vials of varying size and shapes, from a few tenths of milli-liters to several liters, but in most cases we usually deal with plastic bottles of 0.5 - 1 liters. In our labs there can be thousands of different raw materials, each identified with a printed label and a barcode. The fetching of raw materials is handled manually by lab operators, something that takes away a significant portion of valuable time.
The ideal end goal would be to have a mobile robot with a simple pick and place arm that is capable of:

- navigating in the lab and mapping locations of lab benches and raw material racks
- mapping (using vision and/or RFID tags) the location of each raw material autonomously
- respond to dynamic fetching requests from lab operators; these can be pre-scheduled (list of materials to have available on specific date) or live (requested on the spot, to be executed as soon as possible)
- cleaning up function, i.e. browsing lab benches where raw materials are no longer needed and putting them back in the original place

Impact

Business Impacts

The applications presented in this case study have a direct impact on the innovation throughput and time to market of FMCGs research and development organization. The key bottleneck is man-power, especially for all those situations where manual testing is involved. If we enable automatic testing via robots we can exponentially grow the throughput of testing labs, going into 24/7 operations; this alone would increase the throughput by a factor of 4X assuming the same speed as a human operator. Higher throughput also means less time needed to bring new products to the market.

Other Impacts

In our job satisfaction surveys, it is evident that repetitive manual testing negatively affects the morale of employees. These tasks are usually perceived as boring and non creative; by rolling out collaborative robots that can take on this work we would positively impact the wellbeing of our work-force by enabling people to concentrate on more creative and value-added activities.

Measures of Success

Our robotics applications today are essentially hard-coded with very little “intelligence” added. Even basic changes, like for example moving the location of an instrument, or changing the order in which we want to perform certain tasks, require the intervention of a programmer to modify the source code of the procedure. Our measures of success can be described on three levels:

1. Degree of autonomous flexibility - with AI planning we aim at making our current procedures more adaptable to a changing environment and work-process
    a. Autonomous scheduler of operations (minimum goal): the robot unit is capable of autonomously planning the course of action for any combination of samples to be analyzed (number and type) and specific analysis sequence for each sample, within the agreed range (max number of samples and pre-defined number and type of instruments)
    b. Robustness to unexpected events and failures (minimum goal): the robot unit is capable of detecting failures and unexpected events (for example the pouch fell from the gripper) and recover autonomously by taking corrective actions. The overall testing plan is not interrupted (productivity loss).
    c. Adaptability to modified environment (stretch goal): the robot unit can autonomously adapt its operation to an unexpected environment change, for example if an instrument is moved to a different location, or a new obstacle appears in the operating area.

2. Overall efficiency of the robotic procedure - we would like to measure the impact of the AI planner in terms of overall productivity. We will test in parallel the best possible prototype developed by an expert programmer with sequential coding and compare it with the version that uses the UPF to achieve the same task. We expect a gain of productivity of at least 30%, measured in terms of time it takes to complete the assigned procedure at parity output.
3. Enlarge the user base - Today we only have a handful of people that can program advanced applications or modify existing procedures. As a measure of success of the UPF we will report how many non-expert people will be enabled to adapt and modify existing applications to cope with new needs, something that would have required in the past the effort of a specialized robot programmer.

Planning Integration

Our lab robots are currently not integrated with our bigger corporate systems today, except for the upload of test results in our databases; they will therefore be ideal test beds for the application of the UPF. We are going to build a dedicated development environment for testing the UPF application, an isolated machine connected with the robot and the various instruments.

Upon successful validation we will integrate the unit with:

- Corporate request and scheduling system - test requests and samples data will be sent directly to the robot unit and the planner
- Export of results and data - the produced data and sample identifier will be saved automatically to the corporate system for analysis and activation

Requirements

AE-010 – Sequential Plan Generation	Type: **F**	Verif: **T**
The UP shall allow the automatic generation of sequential plans in P&G lab		
Comment: The P&G automated experiment domain describes all actions to be performed by a robotic arm and measurement devices to provide several measures about consumer goods quality.		
Testing procedure: Given as input the description of the experiments and of robot's and devices' actions, ask the UP to automatically generate a sequential plan to be executed to obtain the expected results.		
Evaluation criteria: The plan is generated, it is valid and satisfies the specified goals.		

AE-020 – Optimal Sequential Plan Generation	Type: **F**	Verif: **T**
The UP shall allow the automatic generation of optimal sequential plans (with respect to plan length) in P&G lab		
Comment: The P&G automated experiment domain describes all actions to be performed by a robotic arm and measurement devices to provide several measures about consumer goods quality.		

	Testing procedure: Given as input the description of the experiments and of robot's and devices' actions, ask the UP to automatically generate an optimal sequential plan to be executed to obtain the expected results.
	Evaluation criteria: The plan is generated, it is valid, satisfies the specified goals and it is optimal with respect to plan length.

AE-030 – Optimal Sequential Plan Generation with Actions Costs		Type: **F**	Verif: **T**
	The UP shall allow the automatic generation of optimal sequential plans (with respect to action costs) in P&G lab		
	Comment: The P&G automated experiment domain describes all actions to be performed by a robotic arm and measurement devices to provide several measures about consumer goods quality.		
	Testing procedure: Given as input the description of the experiments and of robot's and devices' actions an estimation of actions costs (in terms of execution time), ask the UP to automatically generate an optimal sequential plan (in terms of total execution time) to be executed to obtain the expected results.		
	Evaluation criteria: The plan is generated, it is valid, satisfies the specified goals and it is optimal with respect to total action cost.		

AE-040 – Parallel Plan Generation		Type: **F**	Verif: **T**
	The UP shall allow the automatic generation of parallel plans in P&G lab, taking into account the possibility of moving the robot arm during the measurement operations.		
	Comment: The P&G automated experiment domain describes all actions to be performed by a robotic arm and measurement devices (modeled as multiple agents) to provide several measures about consumer goods quality with some degree of parallelism.		
	Testing procedure: Given as input the description of the experiments and of robot's and devices' actions, ask the UP to automatically generate a parallel plan to be executed to obtain the expected results.		
	Evaluation criteria: The plan is generated, it is valid, satisfies the specified goals, and its execution is faster when compared with sequential plans, due to the presence of parallel actions.		

AE-050 – Planning time		Type: **P**	Verif: **?**
	The UP shall allow the automatic generation of plans in P&G lab with reasonable planning time.		

	Comment: The UP planners should return plans in short time, although the process is expected to be performed off-line with respect to plan execution, so planning time would not be a critical factor.
	Testing procedure: Given as input the description of the experiments and of robot's and devices' actions, ask the UP to automatically generate plans comparing different planners.
	Evaluation criteria: Plans are generated by different planners and planning time and quality of the solutions are compared.

T2.7 Subsea robotics

Underwater operations in the O&G industry include inspection and maintenance of pipelines, wells and complex underwater processing systems. The state of the art is largely based on remotely-controlled unmanned underwater vehicles. This solution is extremely costly since the underwater vehicle needs to be connected by wire to an on-surface control vessel.

HyDrone is a technology program to develop new concept underwater vehicles, highly reconfigurable with different payloads, expected to carry out underwater missions at different degrees of autonomy and to operate resident underwater without maintenance for up to two years. Typical missions range from geographically restricted inspections and operations on wells, pipelines and processing systems (using wired communication and power), to medium range missions (based on underwater wifi communication and battery power), to long-range missions.

AI planning techniques may help to overcome several problems like:

- Management of system resources even in abnormal or unexpected conditions happening during mission execution
- Management of critical/abnormal situations that may lead to damage or risks in general
- Making the mission’s definition independent from specific field, so allowing to develop a much more general code in the lowest levels of the architecture (improving reusability, testability and development of additional features for the system and not for a specific scenario)
- Allow planning missions that foresee use of manipulator and tools in general in a seamless way with the rest of the mission
- Exploiting opportunities that may not be exploited by standard, rigid plans programmed by humans
- Help in validating missions proposed/developed by humans

This document describes Saipem’s use case, trying to highlight all the requirements that an autonomous planner should full-fill to extend at its maximum the technology’s added value.

Context

Underwater operations in the energy industry include inspection and maintenance of pipelines, wells and complex underwater processing systems. A typical state of the art situation is largely based on remotely controlled unmanned underwater vehicles as shown in Figure 1. This solution is extremely expensive, since the underwater vehicle needs to be connected by wire to an on-surface control vessel, which needs to have human specialists and pilots on-board to be correctly operated.

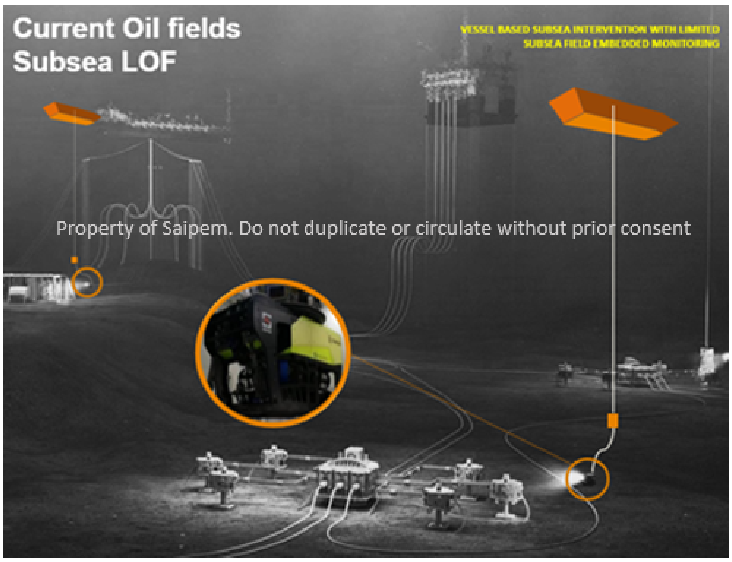


Figure 1: state-of-the-art situation of the inspection and maintenance of underwater oil fields.

HyDrone is a technology program of Saipem aimed at developing new-concept underwater vehicles (see Figure 2), highly reconfigurable with different payloads, expected to carry out underwater missions at different degrees of autonomy and to operate resident underwater without maintenance for up to two years.

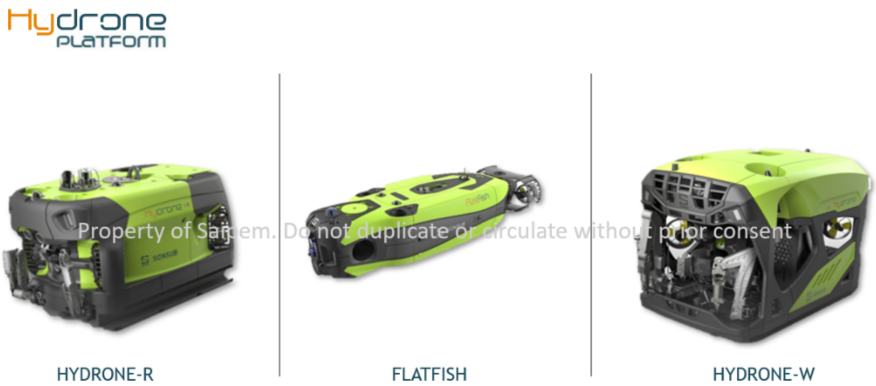


Figure 2: Vehicles of the HyDrone technology program

Typical missions range from geographically restricted inspections and operations on wells, pipelines and processing systems (using wired communication and power), to medium range missions (based on underwater Wi-Fi communication and battery power), to long-range missions, such as inspecting long pipelines from two extraction wells or inspecting subsea installations far away from vehicle base.

Long term missions are to be carried out in full autonomy, in battery-operated mode, and must be fulfilled without any form of communication. In this scenario, AI planning techniques may help to overcome several problems, as detailed in the following Sections.

Planning Application

Planning technology is of paramount importance for the Hydrone platform program. Indeed, as in any other autonomous robotics context, planning is a must-have feature to prevent failures and overcome the limited a-priori programming the human mind can give to the robot.

Without planning, the only way the Hydrone platform would have to carry-out its goals, would be to rely on solely human-written missions. Those missions could bring the vehicle in dangerous situations, such as, but not limited to, not feasible paths with the current vehicle energy, collide with obstacles, damaging expensive tools, or wasting time and money with pointless operations.

The biggest consequences of such events are multiple: (1) the vehicle will go to an emergency state to prevent collisions or not to finish its resources; (2) the vehicle can be lost and (3) potential damages to clients' structures (due to failure in intervention).

While it is clear that (2) would be a loss in the millions magnitude order, even (1) is not auspicable, since each time the vehicle goes to emergency, it powers off its sensors and enables the emergency pingers to be retrieved. The cost of such retrievals, though, are quite expensive, since a fully operated vessel, which costs easily over 100k€ per day, needs to go where the vehicle is supposed to be found. (3) is very difficult to quantify, due to extreme variability of potential impact on productivity of client's subsea energy field and associated reputational damages for Saipem.

Thus, the aspects that could be improved by using planning are the following:

- **Resource management**. Principal resources for underwater vehicles are energy (battery level); time constraints; memory (data storage level) and computational power (CPU usage). For example: a multitarget mission may be planned in different ways, achieving the same goal(s), but consuming different amounts of resources depending on the environmental conditions, time elapsed to complete the mission, preferences and priorities given on resource consumption, etc. Alternatively, in case of an anomaly detected during a pipeline inspection the capacity to re-plan the mission and plan a deeper inspection taking into account residual battery level, data storage availability and time required to complete the survey before informing field operators of the anomaly (which may be critical depending on the consequences of the detected anomaly).

- **Abnormal conditions and failure of mission execution**. Navigating in a complex scenario complicates the a-priori modelling of the possible faults and abnormal conditions the vehicle may encounter. However, unexpected system or environment conditions may lead to critical consequences, such as loss of vehicle, service unavailability for client or environmental damages. This is something the planner should consider and manage, while a mission is performed. For example, if the time to complete a mission is not meeting certain deadlines or an unexpected obstacle is detected, the planner should intervene, re-planning the mission by taking into account the mutated system status and overall conditions, to ensure the mission completion or the reach of a safe location.
- **Mission definition independent from the field**. Missions, the definition of paths and trajectories, are commonly field-specific: they can be slightly parametrized at best. Safely configure underwater vehicles for different fields by just providing the field map and the vehicle model (or relevant parameters characterizing the vehicle) and features would noticeably improve system effectiveness. In fact, it would allow operators to define missions through simple HMIs, by just providing mission targets and their priorities. For example, the goal "open valve located in XYZ location" is field-specific, while an autonomous planner should use a set of instructions goal-oriented and re-usable in any field such as "open valve VALVE-IDENTIFIER ID".
- **Autonomous manipulation and tools usage**. All the HyDrone family vehicles will have the possibility to use/interface with tools fitted on dedicated skids. Indeed, the first vehicle produced, the HyDrone-R, is also equipped with a robotic manipulator arm, in order to carry-out light manipulation tasks, such as operating a valve. Planning techniques are thus required in order to solve the motion planning of the manipulator, considering a possible future autonomous manipulation. Motion planning should also be integrated in the general mission planning: the target is to have a multi-domain planning capacity, completely transparent for the operators.
- **Exploiting opportunities**. An autonomous planner should be able to take advantage of any opportunity. Indeed, many events may happen while executing a mission. For example, a typical requirement for a resident vehicle may be to carry-out periodical inspections on specific items placed in the field. The a-priori knowledge of the inspection deadlines could be used to add goals to other missions when this represents the opportunity to match the periodical inspection schedule without need of a dedicated mission. On top of this, it is expected that the planner, knowing the status of the system and comparing it to the system status forecast, done during planning, can add or remove ancillary targets from the mission, or adjust the mission to deal with unexpected/unplanned conditions.
- **Validation of missions proposed by humans**: While most advantages about the planning capabilities are experienced with the vehicle in autonomous mode, it shall also be possible to ask the vehicle to perform a mission completely written by a human. Planning here is useful to validate the inserted mission, to prevent faults such as: battery problems, disk space problems or prohibited configurations before they happen.

To fulfill the aforementioned points, the planner needs to have access to internal states and description of the vehicle. In particular, the following:

- **The vehicle state**. Booleans such as, low battery, uncontrollability of the vehicle, sonar not working. Other types of variables included in the state could be the remaining level of battery, the free disk space, the CPU usage. These values are filled by the lower levels of the vehicle middleware and made available to the planner.
- **Model of the vehicle**. The model contains map and functions of how the vehicle resources vary, e.g. the battery discharge model. Other than that, the model contains even the details of the vehicle such as the actions it can perform (most lines of the mission plan would be a start/stop of them). Possible additional

target is the capability of the planner to improve the model, collecting and analyzing data mission after mission.

- **Map of the scene**. At each moment, the planner needs to have a clear picture of a semantic map of the scene where the vehicle is supposed to operate. This information is static and not supposed to change with new decisions.
- **Goal list with priorities**: The input for the planner.
- **General constraints or constraints on vehicle status**: as an example, a custom constrain to forbid the vehicle to cross a certain area which is not in the exclusion zones of the semantic map.

The schema that maps the interaction between the UPF and the Hydrone platform is depicted in Figure 3. Hydrone platform vehicle control systems are composed of hundreds software components stratified in a complex network interaction. For the sake of planning, the only Hydrone component which plays an active role is the Orchestrator. The Orchestrator is a software module that, provided a mission file, it sends to the lower levels the needed commands to execute them in the right order and taking care to follow the desired schedule. Moreover, it is also able to trigger simple predefined mission files to respond to certain abnormal conditions.

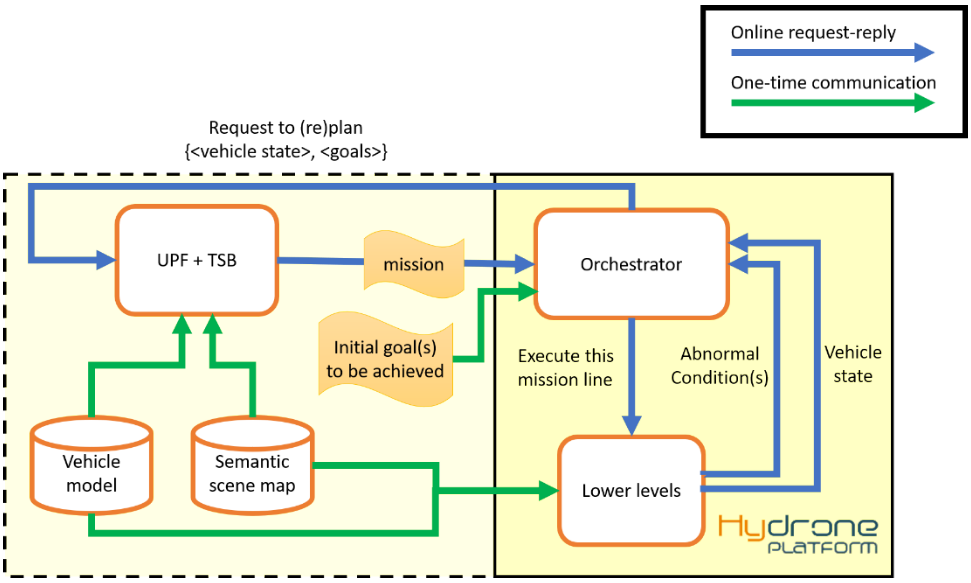


Figure 3 Communication between the UPF and the Hydrone platform vehicle.

As the careful reader can understand, the Orchestrator is, thus, the natural choice to map an interaction between the UPF to request a planning (or replanning) in order to obtain a new mission file updated with the state of the current mission and environment. Figure 3 shows that the UPF will need to also understand the vehicle model and the semantic scene map, as those will be needed input for the planner.
It is foreseen that the Orchestrator would need to ask the UPF a plan/replan in the following cases:

- **New mission request**: This is the case where a user/pilot asks the Orchestrator of the Hydrone vehicle to reach a certain goal(s), considering a list of constraints and, possibly, priorities among the goals. Since no mission is directly provided by the user, but just a set of goals, the Orchestrator needs to ask the UPF to perform a mission by means of planning. The frequency of this type of request is expected to be not more than one per minute.
- **Abnormal condition occurs**: Without planning, this point could be solved by pre-defined mission files provided to the Orchestrator to be started in case of abnormal conditions. If we have the UPF, though, it should be possible to ask it to replan to obtain a new mission file adapted to the change of vehicle state/environment. The frequency of this type of requests is expected to be not more than one per minute.

- **Asynchronous polling-based updates**: It should be possible to give the UPF/Planner the possibility to change the current mission execution, due to, as an example, new opportunities met in the field. Thus, the Orchestrator will communicate the current state of the vehicle based on a certain polling rate. This type of communication is expected to be one per polling rate, where the required polling rate should not be lower than 15 seconds, neither longer than 1 minute. Thus, it is expected the UPF to be able to receive updates even in such case. It is not required that the planner reacts at the same rate, nevertheless it shall be possible to force the planner to answer within a required time-period, even if with sub-optimal planning solution, or with a "no solution found" reply.

It is also foreseen that the UPF could be installed locally inside the vehicle control system. This way, the communication latency between the Orchestrator and the UPF would be as less as possible.
An additional, long term, development scenario may be the introduction of fleets/flocks of Hydrone vehicles (collaborating to achieve overall goals) and/or being supported/launched/recovered by autonomous vessels, making the mission much wider, complex and with targets split among the different type of vehicles.Having this in mind, designing the UPF (and its APIs) in a way that make it open to deal with this kind of scenario in the future will be extremely useful (nice to have requirement).

Introductory Example

Example: Operator needs to send Hydrone vehicle to close 2 valves in a defined order, to allow performing scheduled maintenance activities on the wellhead.

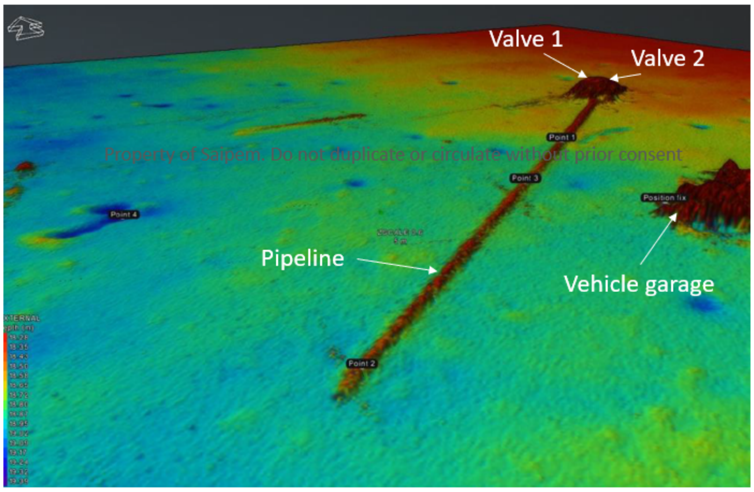


Figure 4 - The digital map of the oil field where the Hydrone vehicle needs to operate.

The operator communicates his needs to the Orchestrator using the provided HMI. The Orchestrator then requires the UPF to provide a mission file for the goals it received. Since the UPF knows the map of the scene, the model of the vehicle and the current vehicle state, it is possible for him to plan a mission file.
The Orchestrator then receives the mission file and begins to execute it by informing the lower levels of the vehicle. Each polling rate, it gets the vehicle status and abnormal conditions to understand if the mission is being followed properly.
In case of critical abnormal conditions, such as, critical battery level or similar, the Orchestrator stops immediately the mission execution and proceeds to perform a safety countermeasure by invoking a predefined mission file. On the other hand, if the abnormal condition is not critical, such as an unexpected obstacle not avoidable in an easy

manner, or abnormal resources consumption (different from what forecasted), it communicates the event to the UPF. If it receives a new replanned mission file, it executes it, otherwise, it decides whether to perform a safety countermeasure or continuing the previous mission.
While the mission is performed, the UPF is constantly informed of the new vehicle status, to give to the UPF the possibility to change the mission being executed in case of opportunities, or if the course is not going as the simulated behavior.

Impact

Business Impacts

Section 2. already outlines the list of advantages of the planning integration for autonomous vehicles, such as the Hydrone platform vehicles. Thus, the business impacts can be summarized as the following list (from most to less impacting):

- **Increase business opportunities (enabling technology):** autonomous subsea machines are basically not existing in the offshore energy market; having an advanced planning capability may allow Saipem, as owning company, to offer to its clients services that were simply not possible without this enabling technology
- **Reduce vessel operations**: With advanced planning-enabled resident autonomous vehicles can reduce by a big amount the need for offshore vessel operations that can easily cost over 100k€ per day.
- **Reduce recovery of autonomous vehicles due to emergency**: When a vehicle goes to emergency, it needs to be recovered by a vessel and specialized tools, such as LARS. Planning should give a better degree of autonomy to such vehicles, thus reducing the possibility of an emergency. It may also happen to lose a vehicle that surfaced due to emergency conditions, in such a case potential damage is the cost of the vehicle (new vehicle cost is in the 1 to 5 millions range, depending of vehicle type, payloads included, etc)
- **Reduce damage risks**: An AUV moves inside oil fields rich in expensive tools and structures. Having planning onboard, reduces the risk of uncaught events, thus reducing the risks of damaging such tools and possibly the environment.
- **Reduce training time for operators**: Without planning, an operator should give to the Orchestrator directly the mission file to execute, thus replacing the planner. This would result in having to train operators to be expert enough to provide the best mission plans.
- **Prevent future intervention and failures of subsea assets through continuous opportunities monitoring and replanning.**

Other Impacts

- **Reputational risk/opportunities:** given the novelty of autonomous vehicles with planning capabilities in the offshore energy market there is good reputational opportunity in having a vehicle with this ultra-advanced, with respect to the rest of the market, capabilities.

Measures of Success

Given the high novelty for the sector, the evaluation will be mostly qualitative.
We will simulate and compare missions written by domain experts and missions, with the same goals, produced by the UPF changing resources availability. To consider abnormal conditions and failure management, we will introduce anomalies during the simulated mission execution and verify that the planner triggers re-plan actions and keeps the vehicle safe even when the system or the environment show abnormal conditions. To validate the exploitation of opportunities, a periodical inspection plan will be given to the autonomous planner; In general, we expect that the planner adds goals to missions to match the plan deadlines.
Finally, SAI has a playground for the experimentation of real subsea robots in Trieste: if possible, we will experiment with the same scenarios with real vehicles.

Performance	KPI
Success of planning	● Percentage of missions successfully completed running with planner is higher, also considering: o Abnormal conditions triggered o Unexpected environmental conditions (or unplanned abnormal conditions) ● Planner is able to produce valid missions matching more targets than human even in "lack of resources"
Efficiency of planning	● Planner use less time WRT human to plan a VALID mission ● Planner missions use less resources, according to resource consumption priority required, WRT human planned missions (comparing valid missions)
Exploiting of opportunities	● Missions executed with planner running will adjust targets adding or deleting ancillary (nice to have) targets depending on mission development (real behaviour of vehicle taken into account instead of just vehicle model) e.g. replanning ● Vehicle model will be adjusted learning from "experience": data coming from missions helps in correcting model, after reasonable number of missions planner should become more accurate in predicting resources consumption ● Missions will include/propose ancillary targets, not required by operators, to match the field periodical inspection and maintenance plan (e.g. if there is a periodical inspection to be carried-out in the next future close to the mission's targets required by operator, that will be added to the mission's targets so to avoid making a dedicated mission in short time)
Vehicle safety	● Safety of vehicle is preserved even in case of unexpected-unpredictable abnormal conditions, this is not achievable with human planned missions and no planner running while mission execution is on-going

The planning result given by the UPF should give a clear indication of:

- **Solution found or not**: if the solution has been found or not. In case it is not found, it would be nice to have an indication of the reason why (no solutions for the given problem or not enough time given to the planner to find a solution). Solution found means all mandatory targets accomplished.
- **Percentage of success / goals accomplished by mission plan**: percentage of ancillary targets accomplished by mission
- **Quality of solution** in case of prioritization of resource optimization this is meant to be a measure of effectiveness in resource optimization process
- **Forecasted/Milestone vehicle state**: Once the solution is provided, it should also be provided the forecasted final vehicle state. This way, the Orchestrator could decide to perform the mission or ask for a replanning, relaxing the constraints given before to obtain a better result. Moreover, it could be interesting to also have some milestone vehicle states. This way, the Orchestrator would be able to have a runtime check on the mission execution.

Planning Integration

Since planning will be fully integrated in the machine lifecycle while a mission is performed, it is foreseen that the communication between UPF and the Orchestrator will be as fast as possible.
Other than that, since the UPF will be included in an autonomous machine control system it should be possible to:

- **Shutdown/Demote the UPF:** This measure could be needed to prevent the CPU usage in certain cases trying to respond to a critical abnormal condition. Indeed, in this case the Orchestrator should be able to shutdown the components not needed and take total control of the control system.
- **Give the UPF a maximum timeout for the planning time:** Planning is sometimes a time-demanding task. For the same reasons explained before, the Orchestrator or the actor involved with the request of a plan, should be able to specify a maximum planning time to the UPF. To this end, it would also be a useful feature to have a way to inform the UPF to use a different planning method in case the time is limited. As an example, most times, looking for non-optimal acceptable solutions in a search space is way faster and still providing good results (such as solutions find by meta-heuristics approaches).
- **UPF footprint should be limited:** In order to make the UPF available to be installed in already crowded control systems, the UPF footprint on CPU and RAM usage should be as limited as possible. Acceptable measures are 100% of one Intel-i7 2.6GHz CPU core (or equivalent) and 4GB of RAM.
- **Force the UPF to NOT use GPU;** in case a GPU is needed, it needs to understand the footprint and the possibility to not use it.

Example and Evaluation Data

For the Hydrone world, a mission plan is a file containing a set of instructions. Each instruction can:

- give information regarding a specific, standard operation to be performed on a task. A task is a collection of software libraries and binaries that permits to achieve a specific goal or implements a certain behavior.
- Check on a specific condition
- Wait for a specific time

Considering the problem already cited in Section 3, a possible (human hand-crafted) mission plan that fulfills the requests, could be following:

```
Start(Control());
Start(AutonomousUndocking());
Wait(vehicle_undocked);
Start(Goto(location_of_the wellhead_where_the_valves_are));
Wait(Goto_finished);
Start(LocalizeValve(valve1));
IF valve1_localized THEN
  Start(ApproachToManipulate(valve1));
  Wait(Approach_finished);
  Start(AutonomousManipulation(valve1));
ELSE
  AbortMission();
IF valve2_localized THEN
  Start(ApproachToManipulate(valve2);
  Wait(Approach_finished);
  Start(AutonomousManipulation(valve2);
ELSE
  AbortMission();
Start(Goto(home));
Wait(Goto_finished);
Start(AutonomousDockingAndRecharge());
Wait(vehicle_docked);
```

While the mission plan specified could fulfill the aforementioned set of goals, the result could easily be non-optimal, since there is no feedback and replan in case of abnormal or unexpected events.

As an example, the answer to the question: "what would happen if the vehicle localizes valve1, but then is not able to localize valve2, since the approaching task made it disappear from the field-of-view of the cameras?" Is not clear before a simulation run of the mission. Even in that case, the simulation could fake the water turbidity and the distortion of the real-world lenses, so the real-world result could still be different in such a situation.

For this reason, recalling the KPIs defined in Section 5, we would have that:

- **Success of planning**: a planner could provide a better success rate, indeed, even if it were not able to plan a good mission file in the first instance (since the planner is expected to be not better to simulate than the simulator itself). Indeed, in case of an anomaly like "valve not found" instead of aborting the mission, the planner can generate a new one, trying to exploit a variety of new possibilities, such as, trying to reach the valve from another angle/perspective.
- **Efficiency of planning**: Provided that the aforementioned simple mission is VALID, it took 10 minutes to be written by a human. Considering that no big reasoning is done regarding energy consumption, disk space consumption, temporal planning or task multithreading, the planner should be able to generate a mission for this simple case in a shorter time interval. Moreover, in the general case, the time for a human increases exponentially, while it will remain the same for the planner (since it will always deal with such constraints).
- **Exploiting of opportunities:** What would happen if during the mission there is a damaged third valve in the field-of-view of the cameras while the vehicle is inspecting the two valves? In the hand-crafted case, the operator will know that just by delogging the mission data and possibly triggering another mission specifically written to inspect that event. On the contrary, the planner could anticipate this, always taking into account its constraints regarding the vehicle resources.
- **Vehicle safety:** What would happen if there is an unforeseen dead battery pack while the vehicle is traveling to reach the (possibly far-away) valves to operate? In the hand-crafted solution, the vehicle will trigger an abnormal condition and the Orchestrator will try to save the vehicle, eventually triggering an emergency. On the contrary, the planner could understand better what to do with the remaining energy, for example, trying to reduce the distance from the vehicle to the initial docking point to reduce the recovery difficulties.

The KPIs will be evaluated using the following schema:

We will give to the human expert and to the autonomous planner the task to provide a mission file for the following cases:

- Easy missions:
  - o 1 mandatory goal and 1 optional goal
- Standard missions:
  - o 3 mandatory goals and 3 optional goals
- Difficult mission:
  - o 10 mandatory goals and 5 optional goals

The table summarizes the evaluation:

Performance	**KPI**	**TEST**	**Evaluation criteria**
Success of planning	Percentage of missions successfully completed running with planner is higher, also considering: ● Abnormal conditions triggered ● Unexpected environmental conditions (or	Execute easy, standard and difficult missions planned by human and by UPF, at least: During mission execution simulate triggering of abnormal conditions During mission execution simulate unexpected adverse environmental conditions (e.g. strong water current)	It is expected that: ● In simple missions both human and UPF will have similar scoring in terms of completed/aborted missions. Possibly the UPF manages better the unexpected environmental conditions

	unplanned abnormal conditions)	During mission execution simulate both abnormal condition and adverse environmental conditions (repeat triggering in different order, triggering different events and triggering at different mission progress)	● In standard missions UPF should have better scoring and will be able to complete more missions under any circumstance. Number of total targets accomplished, even when both completes the mission should be greater when using UPF ● In Difficult missions UPF should have a noticeable advantage in terms of both: number of successful missions and number of ancillary targets accomplished on top of mandatories.
Success of planning	Planner is able to produce valid missions matching more targets than human even in "lack of resources"	Pick the same easy, standard and difficult missions above described, and let programmer and UPD plan them considering, at least, following cases: ● Full resources (e.g. 100% of energy stored in batteries; full storage available; no time constraints) ● 75% of one or more resources available ● 50% of one or more resources available ● 25% of one or more resources available Let programmer and planner repeat the planning giving, additionally, priorities among resources (e.g. priority is to minimize battery consumption) Try to trigger unexpected environmental conditions or abnormal conditions during mission execution	It is expected that UPF will match more targets with respect to humans in the original plan. During mission execution gap should enlarge when abnormal conditions or adverse environmental conditions are triggered during mission execution
Efficiency of planning	Planner use less time WRT human to plan a VALID mission	Pick again the three types of missions: easy, standard and difficult. Measure the time that the UPF and the human programmes take to plan the missions written for the "success of planning" KPIs.	It is expected that the planner will take considerably less time than a human programmer.

		Try to let the UPF plan the missions limiting the amount of computational resources available for the planner itself (e.g. CPUs and RAM)	
Efficiency of planning	Planner missions use less resources, according to resource consumption priority required, WRT human planned missions (comparing valid missions)	Let the UPF and a human programme to plan the easy, standard, and difficult missions (without ancillary targets) giving priority to some resources (e.g. prioritize minimization of power consumption or time for execution)	It is expected that the UPF will produce plans consuming less resources (according to the system model) than the human programmer. When a resource is prioritized, do the comparison also on the non prioritized resources: it is expected that the planner improves consumption on all the resources (prioritized, with maximum impact, and not prioritized, less noticeable)
Exploiting of opportunities	Missions executed with planner running will adjust targets adding or deleting ancillary (nice to have) targets depending on mission development (real behaviour of vehicle taken into account instead of just vehicle model) e.g. replanning	Execute the difficult mission, in the following conditions: ● With the on-board vehicle UPF active and running ● Without the on-board vehicle UPF active and running Then modify the environmental conditions (make them worse and better on different mission's executions) or force some vehicle resource to be much lower/higher than expected	Verify that the on-line planner removes/reorganizes targets to deal with evolving scenarios. In case of actual conditions better than planned should happen the opposite: planner add targets to the mission to achieve better overall mission result
Exploiting of opportunities	Vehicle model will be adjusted learning from "experience": data coming from missions helps in correcting model, after reasonable number of missions planner should become more accurate in predicting resources consumption	Provide to the planner a system model different from the real (or from the one implemented on the simulator). Execute the same mission at least 10 times. Let the planner re-plan the same mission (same targets, same environmental conditions, same initial level of resources) between one execution and the other.	Verify that the planner adjusts the model after each mission execution (or after a bunch of executions).
Exploiting of opportunities	Missions will include/propose ancillary targets, not required by operators, to match the field periodical inspection and maintenance plan (e.g. if there is a	Provide the planner with a list of activities to be performed periodically in defined locations of the field (at least one activity) Give to the UPF a list of targets for a new mission to be done in the area where the periodical maintenance is foreseen.	Verify that the UPF adds targets to the mandatory mission targets to fulfill periodical maintenance tasks, whenever the area proximity and level of available resources allow it to complete it, taking the occasion.

	periodical inspection to be carried-out in the next future close to the mission's targets required by operator, that will be added to the mission's targets so to avoid making a dedicated mission in short time)	Give to the UPF a list of targets for a new mission far from the area where the periodical maintenance is foreseen. Let the planner run with different available resources.	
Vehicle safety	Safety of vehicle is preserved even in case of unexpected-unpredictable abnormal conditions, this is not achievable with human planned missions and no planner running while mission execution is on-going	Execute a planned mission, possibly standard or difficult level, having the online UPF active and running in the vehicle. During execution simulate some unpredictable condition (e.g. adverse environmental conditions or sudden drop of battery level). Repeat the experiment without the on-line UPF up and running.	It is expected that with the UPF the system may fail in accomplishing the missions under extremely demanding/unpredictable situations, but will have success in keeping the vehicle safe (going back to garage), avoiding to trigger an emergency float. This should become extremely evident when comparing the results against the same missions executed without the UPF on-line

Requirements

SR-010 – Automatic Plan Generation	Type: **F**	Verif: **T**
The UP shall allow the automatic generation of HyDrone plans		
Comment: The HyDrone domain enclose all actions a HyDrone robot needs to perform to inspect an underwater arena (i.e., reach a set of waypoints in a map while optimizing available resources).		

SR-020 - Plan Validation	Type: **F**	Verif: **T**
The UP shall allow the validation of a HyDrone plan. If the plan is invalid, reasons should be reported.		
Comment: This will allow the operators to check the validity of missions before putting them into operations or simulate them.		

SR-030 - Plan and Optimize Resources	Type: F	Verif: **T**
The UP shall be able to consider (and eventually optimize) a set of input resources (e.g, battery charge, data capacity, moving consumption) while planning to achieve the assigned goals.		
Comment: The generated HyDrone plan should not consume more than the available resources.		

SR-040 – UP shall accept mandatory and optional goals	Type: **F**	Verif: **T**
The UP shall accept mandatory and optional goals. The generated HyDrone plans shall meet all the mandatory goals, while trying to meet even the optional ones (without compromising the vehicle safeness while doing that).		
Comment: If no viable solution is available (i.e., it is not possible to meet all the mandatory goals) the planner shall reply with a clear indication of this.		

SR-050 – Success of planning - UP shall perform better than humans in presence of abnormal conditions.	Type: **PE**	Verif: **T**
It is expected that, when challenged to solve the same problems, the UP will perform better than the human in the presence of abnormal conditions.		
Comment: The percentage of missions successfully completed will consider both the number of triggered abnormal conditions and unexpected environmental conditions (or unplanned abnormal conditions).		

SR-060 – Success of planning - UP shall produce more valid missions than humans.	Type: **PE**	Verif: **T**
It is expected that, when challenged to solve the same problems, the UP will perform better than the human in terms of valid missions produced.		
Comment: The planner is able to produce valid missions matching more targets than human even in "lack of resources".		

SR-070 – Efficiency of planning - UP is more time-efficient than humans	Type: **F**	Verif: **T**
The UP is required to produce valid HyDrone plans in less time than the human, when challenged with the same problem.		
Comment: Planner uses less time with respect to the human to produce a valid HyDrone plan.		

SR-080 – Efficiency of planning - UP is more resource-efficient than humans	Type: **F**	Verif: **T**
The UP is required to produce valid HyDrone plans consuming less resources than a human. This is valid even when challenged with a consumption-priority list inside the problem.		
Comment: Generated HyDrone plans use less resources, according to resource consumption priority required, with respect to the human planned missions (comparing valid missions).		

SR-090 – Exploiting of opportunities - UP is required to replan in case of opportunities	Type: **F**	Verif: **T**
The UP needs to be able to replan an existing HyDrone plan in case of opportunities by adjusting the targets and by preserving the vehicle resources.		
Comment: This will allow adaptation of the plan depending on contingent conditions.		

SR-100 – Exploiting of opportunities - TSB shall enrich current mission with periodic requirements	Type: **F**	Verif: **T**
The TSB shall adapt to plan requests by enriching its output with periodic constraints, identified as new opportunities, such as inspections that need to be performed from time to time.		
Comment: Missions will include/propose ancillary targets - not required by operators and identified as new opportunities of the current assignment - to match the field periodic inspection and maintenance plan. E.g., suppose there is an inspection to be carried out in the next future and, while executing the current assignment, the robot gets closer to the future targets. Such targets will be considered new opportunities for the in-progress mission and will be evaluated for inspection. Fulfilling these opportunities will increase the total reward (in terms of time-saving and resource consumption) of the autonomous vehicle.		

SR-110 – Vehicle Safety - TSB shall preserve vehicle safety	Type: **F**	Verif: **T**
Safety of vehicles is preserved even in case of unexpected-unpredictable abnormal conditions. This feature is not achievable with human planned missions and no planner running while mission execution is on-going.		
Comment: The adaptability of planning is needed to enhance the safety of the vehicle.		

4. Project Demonstrators

In this section, we outline the scope and the expected demonstrations that will be developed during the rest of the project. For each use-case we outline both the level of integration that will be achieved, the TRL and the expected KPI values. Moreover, for each demonstrator we present the expected evaluation criteria and experiments we intend to perform to validate the success of the planning integration.

T2.1 Space domain

For the Space domain the expected demonstrations include two aspects:

- **Automatic Planning** of tactical (one sol) rover planetary exploration operations. Based on the TRASYS 3DROCS tool dedicated for robotic operations specification, monitoring and control, the operator specifies the objectives to be achieved during the given sol and requests from the UP framework the automatic generation of the Activity Plan to reach them. The generated plan is acquired and appropriately visualized on the 3DROCS MMIs. Finally, the plan is validated by the ROSEX operational simulator.
- **Automatic consolidation of Activity Plans**: given a set of candidate Activity Plans prepared for a given sol, the objective is to construct a consolidated valid Activity Plan to be finally uploaded for execution. As in the previous demonstration the candidate Activity Plans are specified by the 3DROCS tool (using the Automatic Planning functionality) and submitted for consolidation to the UP framework. The resulting consolidated Activity Plan is acquired and visualized on the 3DROCS MMIs. It can be afterwards validated by the ROSEX operational simulator.

The evaluation criteria, in order of priority, are:

- Improvement of the time required by the planning team to generate a valid Activity Plan and a valid consolidated Activity Plan. An improvement of 10% (TBC) is proposed as KPI knowing that the operations preparation time between two ‘rover communications’ is very limited. This evaluation criterion is predominant and fully justifies the approach even if no improvement is detected at the following criteria.
- Increase of the number of objectives satisfied by the automatically generated consolidated Activity Plan. A KPI of 10% is proposed as a tentative objective.
- Improvement of the resources usage (duration, power, memory mass) to achieve the given objectives. A KPI of 5% (TBC) is proposed knowing that the nature of the operations is such that it does not provide a big margin of alternatives.
- Explainability of the resulting plans: the presence of explanations when a plan is rejected or a summary report of the satisfied objectives is a desirable objective.

T2.2 Agriculture domain

For the agriculture domain the expected demonstrations include the following aspects:

- **Automatic planning of a silage maize harvesting campaign.** The campaign in the demonstrator will consist of at least four fields and the yield will be stored in at least one silo. The chosen fields will be sufficiently spaced apart so that they will be reached by the harvesters via different routes. For the evaluation, it will not be possible in the scope of this project to use the system already for the coordination of a live harvest. One reason for this is that this would require dedicated test fields, for which no resources are available in the project. Secondly, robust integration on real agricultural machinery is very complex. Therefore, the evaluation will be based on recorded data and a simulation. The campaign planning system will be integrated with a special purpose planner that plans the routes of the machines on the fields.
- **Execution and monitoring** of the plan: Based on the recorded harvesting data we will simulate the execution of the campaign. That basic simulation will vary the travel times of the transport vehicles on the roads, the actual harvest volumes on the fields as well as the compaction time in the silo. To deal with these uncertainties, the system will monitor the plans and make adjustments if necessary.

For evaluation, the following criteria are relevant:
- Improvement of the overall monetary cost of the plan. The plan costs are calculated based on estimates of the actual costs for the individual harvesting operations and labor costs.
- It will be tested if the system is suitable for monitoring of the plans and if it can react to simulated deviations. As an important aspect of this, the runtime needed for planning adjustments should not hinder the harvest process itself.
- Stakeholders and experts in the agriculture domain will be asked for a qualitative assessment of the resulting plans and the overall system.

T2.3 Flexible manufacturing

For the flexible manufacturing domain, the expected demonstrations will include the following aspects:
- **Automated planning & scheduling of gear and pinion**. The system will receive the batch details in a suitable format (extracted from the ASCP report) and using the domain knowledge data, the planning reports will be produced. Human planners will be asked to choose the best optimal plan based on the provided inputs. In case of any demand/ priority change, the system will recalculate the plan and the human planner will be asked to confirm

For evaluation, the following criteria are relevant:
- Delivery commitment and work in progress inventory will be measured against the baseline
- Number of setup changes and the utilization of tool life will be part of the continuous tracking vs today
- System must ensure that the resources are being utilized to the maximum capacity
- The overall result will be measured in terms of delivery and the burden cost

T2.4 Logistic automation

For the logistic automation domain, the expected demonstrations include the following aspects:
- **Semantic check of behavior trees using automated planning.** Through a web based editor developed by Magazino, we will define a set of test behavior trees. The behavior trees will have a JSON representation that will be used as input to a specifically designed technology bridge, which will be used to call the AIPlan4EU planning technology to perform semantic checks. As a result of this activity, we plan on producing a stand-alone Jupyter notebook that, starting from a JSON behavior tree, calls a planner and outputs a list of possible issues.
- **Online planning of robot actions.** For the online planning part, we will replace one of the hand-made trees defining the behavior of the robot in a specific task, with a tree that exploits the planning technology provided by this project.

For evaluation, the following criteria are relevant:
- For the semantic check, we will use the aforementioned behavior trees as input for the semantic check planning problem. Through such examples, we aim at having a proof of concept of how the technology can be used to support users in designing behavior trees through the editor developed at Magazino.
- Behavior tree experts will be asked for a qualitative assessment of the semantic checks and the errors identified by the planning technology.
- In the online planning part, we will produce a sub-tree that leverages the planning techniques of this project. This tree is supposed to have a lower complexity with respect to the hand-made one currently used by Magazino. By counting the number of nodes needed to have an equivalent behavior, we want to demonstrate that this technology can lower the maintenance cost of such trees.
- Stakeholders and experts in the logistic domain will be asked for a qualitative assessment of the plans representing the same robot behaviors.

T2.5 Shuttle fleet management

We will not demonstrate the shuttle fleet management use-case, we will only take into account its requirements for the technical work-packages due to the termination of the EasyMile participation in the project from 01/01/2022.

T2.6 Automated experiment design

For the automatic experiment application, the expected demonstrations include the following aspects:

- **Plan generation and execution of robot behaviors to perform automated experiments.** By modelling the application domain, we will show the use of AIPlan4EU planning technology to generate and execute plans optimizing the P&G quality control experiments in several situations and conditions.
- **Multi-robot coordination.** The second demonstration will extend the first one, by considering multiple robots operating in a coordinated way in the same environment.
- **Human-robot interaction (HRI)**. The third demonstration will show how robots can properly interact with human operators in the environment in order to increase robustness and effectiveness of performed automated behaviors, as well as increasing acceptability and trustworthiness of the AI & Robotics technology.

For evaluation, the following criteria are relevant:

- For the generation and execution of automated quality control experiments, we will measure task performance in relation with modeling effort by non-expert operators and compare the performance of the novel solution with existing hand-crafted behaviors.
- For the multi-robot experiment, we will measure the increase of performance with the number of robots and size of the environment and of the problem.
- For the HRI experiment, we will measure the increase of robustness with respect to fully automated processes and we will conduct a simple user evaluation involving P&G operators to assess acceptability and trustworthiness of the system.

T2.7 Subsea robotics

Subsea robotics planning capabilities will be demonstrated involving the vehicle in a simulated environment, by means of the Saipem simulator (based on Gazebo physics engine). This way, the control chain will be the same one used with the real vehicle and the (re)planning can be tested in a safe manner (hardware in the loop).

Planning demonstration will compare autonomous planner KPI with those of a developer, who will try to do the same job. The missions to generate will be of 3 difficulty levels, easy, medium and hard. For further information, refer to Example and Evaluation Data subsection of the Subsea Robotics Use Case.

5. Conclusion

In this deliverable, we reported the detailed description of the use-cases elicited by the partners of the consortium. Each use-case description follows a common template that was also used for eliciting use-cases from the open-calls programme. Each use-case is described with a level of detail that is sufficient to focus the planning application and its prospective impact. Moreover, for each use-case we defined the expectations on the demonstrators to be realized within work package 5 and we outlined the major KPIs to be evaluated according to the procedures defined in deliverable D6.1 - Evaluation Principles.

This document will be enriched with the use-cases elicited by means of the cascade funding programmes and will become D2.2 - Final Requirements Document at month 30. We also highlight that the use-cases elicited by the project have been published already on the AI4Europe platform and linked to the AIplan4EU project. More information will be reported in these pages following the structure proposed by D7.1 - First Prototype of the UPF Platform Extension and the same will be done with the use-cases elicited by means of open-calls.