Autoware software - Operational Design Domain-based development process

Summary

The Autoware Foundation has adopted an approach to technology development that focuses on a succession of achievable technological milestones. Each milestone is an Operational Design Domain, and each builds on the work of the previous milestones.

The development process is based on two simultaneous loops, one to define ODDs and one to develop support for ODDs in the Autoware software. ODD definition consists of specifying the ODD in unambiguous, but parameterised, terms, and specifying the scenarios that are required to support autonomous driving in that ODD.

The Autoware software is developed to achieve the scenarios for the currently-targeted ODD. In part, verification of the software depends on automated testing of scenarios in simulation, which requires that scenarios are specified in a machine-readable format.

At the completion of support for each ODD, a demonstration is given. These demonstrations serve as stand-ins for applications, showing that Autoware can be used to build autonomous driving applications.

Background

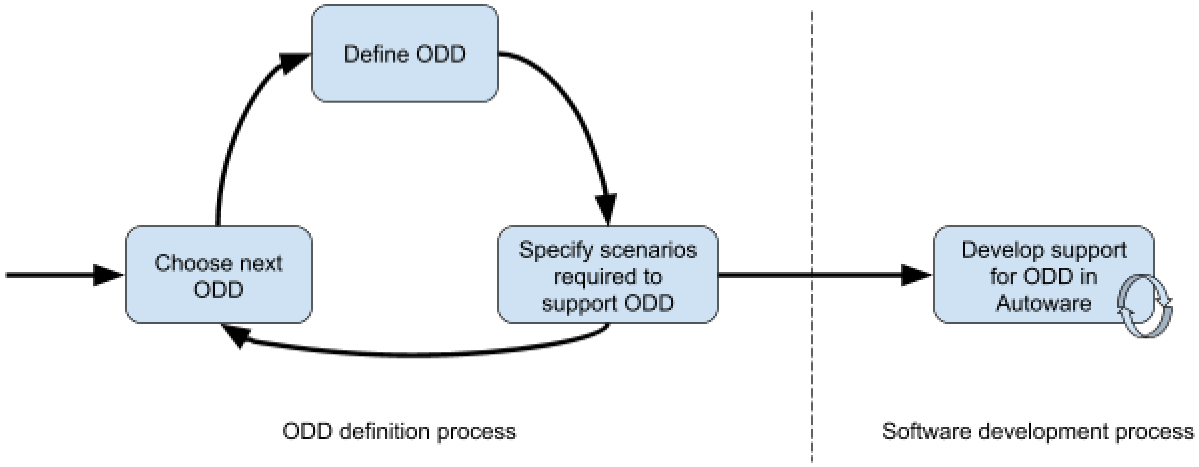
Contributors to the various open-source Autoware software projects come from a variety of backgrounds and have a variety of goals ranging from commercial targets to academic interest.They also have varying attention spans, in that some plan to contribute over a long period of time whilst others operate on a yearly or even shorter basis. Combined with a need to demonstrate a continual progression of development, this creates a need to set common goals for the Autoware Foundation that bring together the disparate contributors and are achievable within a reasonable time frame.

To satisfy this need, the Autoware Foundation has adopted an approach that focuses development work on a succession of increasingly-complex Operational Design Domains (ODDs). The development work is therefore broken down into definition and specification of the next ODD to achieve, followed by software and hardware development to enable autonomous driving in that ODD.

The first ODD targeted by the Foundation is Autonomous Valet Parking.

Overview of the process

At the highest level, the process is two simultaneous loops. One is the process of choosing and defining the Operational Design Domain (ODD) to be targeted next by the software development work. The other loop is the process of developing the support for that ODD in the Autoware software.


Define ODD
Choose next ODD
Specify scenarios required to support ODD
Develop support for ODD in Autoware
ODD definition process
Software development process

This high-level process has several important consequences.

- The Autoware Foundation's development efforts are driven by ODDs, not specific applications or use cases.
- Development efforts will only focus on one ODD at a time.
- Each ODD chosen for development must be a sensible progression from the previous ODD, rather than a large technological leap. (For example, if the current ODD is "cargo delivery in a private, closed road with no people present", the next ODD cannot sensibly be "taxi service in a busy urban environment".)
- Each next ODD should ideally provide some increase in capability over the existing supported ODDs, rather than just being an application of the current Autoware capabilities in a different ODD.

Defining an ODD

Choosing the ODD

The next Operational Design Domain to target is chosen by the Autoware Foundation's TSC. When choosing the next ODD, the TSC must take into consideration a wide range of factors including, but not limited to:

- how big a technological step the ODD is from the currently-supported ODDs,
- the capability of current-day technology to achieve autonomous mobility in that ODD,

- the resources required to achieve that ODD compared with the resources available to the Autoware Foundation,
- the likely required development time (ideally an ODD should not take longer than a year to achieve), and
- the goals of Autoware Foundation member companies.

Defining the ODD

Prior to choosing and defining the scenarios for an ODD, the parameters of that ODD must be defined.

What constitutes an ODD is not yet well-defined in the autonomous driving industry. The definition given by SAE in J3016:2018, where the term was first defined, is:

> Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.

In general, we expect an ODD to be defined by a set of universal parameters that specify the parameters common to all scenarios that may be needed in that ODD.

This definition work falls on the ODD working group, described in the corresponding section below.

When enough information about an ODD is fixed, scenarios for that ODD can be specified.

Specifying scenarios

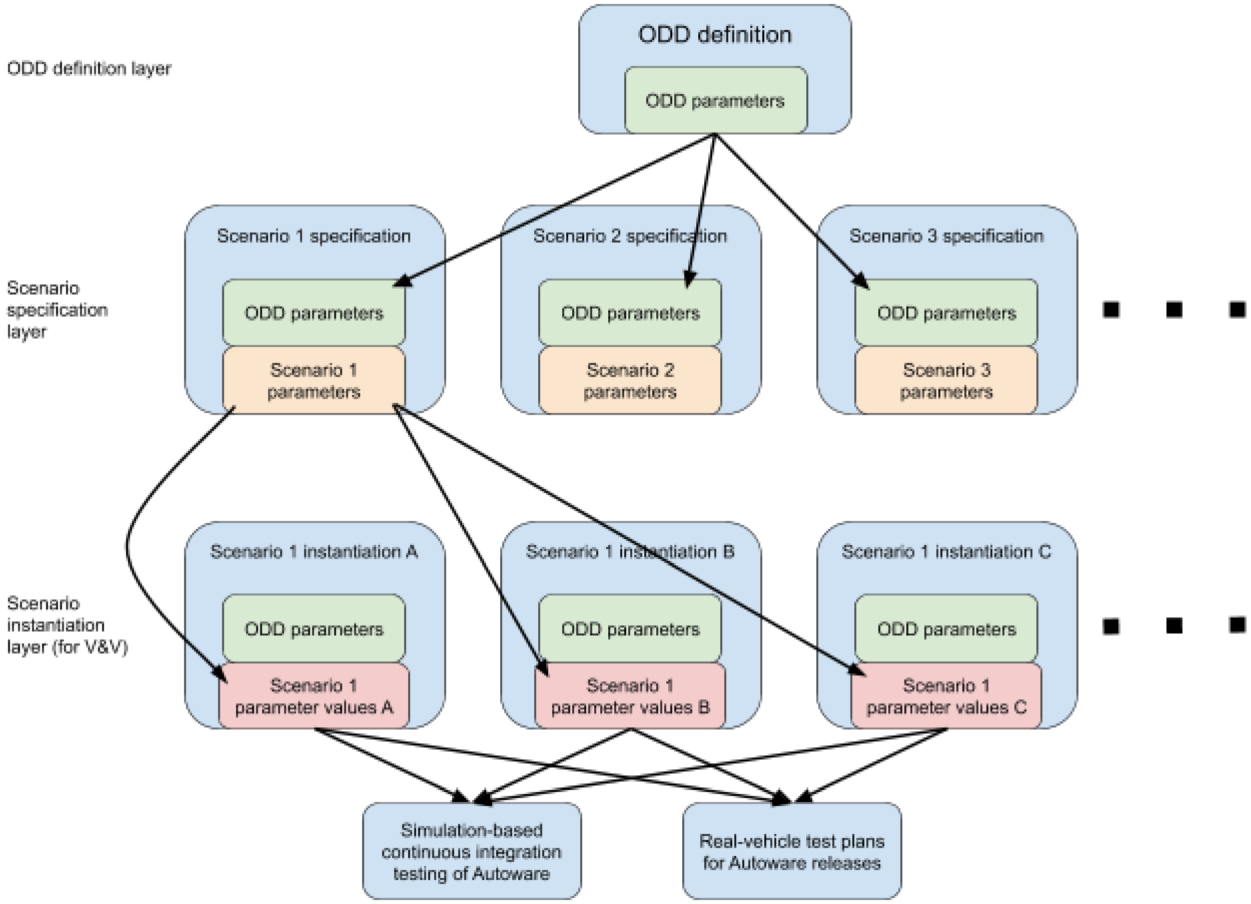
To begin development of support for a new ODD in Autoware, the developers must know what behaviour will be required. The necessary behaviour is specified as scenarios. Each ODD must define a list of required scenarios. This list will therefore fully specify the necessary capabilities of Autoware for that ODD.

A scenario is a specific manoeuvre that the ego car should perform combined with the environment in which the ego car must perform that manoeuvre. A manoeuvre could be something as simple as driving straight in our own lane along a straight road with no road signs, active traffic rule changes, and other cars, at midday on a sunny day. Or it could be something as complex as driving through the Magic Roundabout in Swindon in heavy rain at night during rush hour.

For Autoware, we will use *parameterised* scenarios. This means that each “template” scenario can be varied by altering a range of parameters, such as the weather, the time of day, the colour of a specific non-ego car featured in the scenario, and even specific event variations such as whether a lane-change scenario features a child chasing a ball into the road or not. The version of a parameterised scenario with specific parameters (the instantiated scenario) becomes one test case.

It is important that scenarios be defined in a machine-readable format. This will enable, for example, automated testing in a CI system.

The below diagram shows how the ODD definition, scenarios, parameters, and Autoware interact.


ODD definition layer
ODD definition
ODD parameters
Scenario specification layer
Scenario 1 specification
ODD parameters
Scenario 1 parameters
Scenario 2 specification
ODD parameters
Scenario 2 parameters
Scenario 3 specification
ODD parameters
Scenario 3 parameters
Scenario instantiation layer (for V&V)
Scenario 1 instantiation A
ODD parameters
Scenario 1 parameter values A
Scenario 1 instantiation B
ODD parameters
Scenario 1 parameter values B
Scenario 1 instantiation C
ODD parameters
Scenario 1 parameter values C
Simulation-based continuous integration testing of Autoware
Real-vehicle test plans for Autoware releases

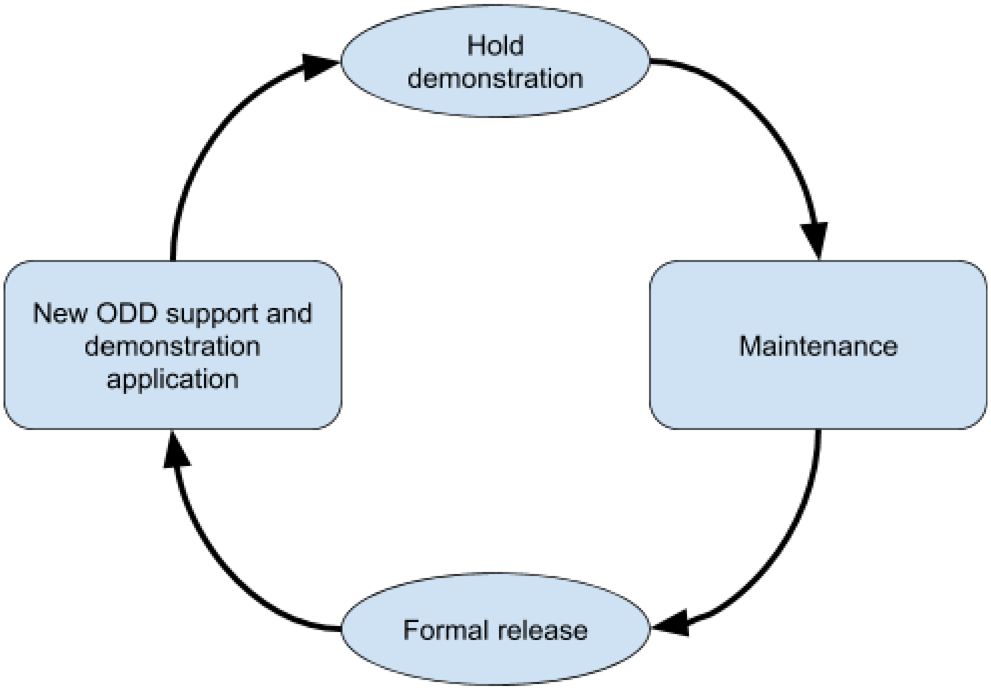
Achieving support for an ODD: The software development process

The software development process is a continuous cycle between two goals.

1. **ODD goal**: Achieving the capability to autonomously drive in a new Operational Design Domain using only Autoware Core, along with a demonstration of that capability through a simple demonstration application.
2. **Maintenance goal**: Cleaning up the project, which includes reducing technical debt, refactoring for improved maintainability, improving documentation, creating tutorials, etc.,

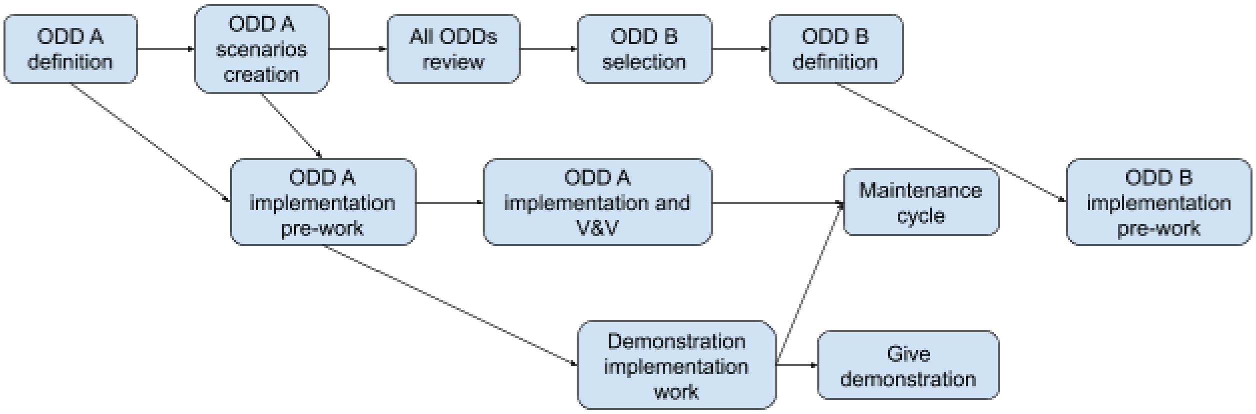
with the aim of producing a formal release of Autoware Core, and optionally Autoware Universe, that supports the newest ODD in addition to all previously supported ODDs.

The reason for using such a process is it allows us to accept some technical debt in the process of achieving a new ODD (and its accompanying demonstration) rapidly, while also ensuring that time is allocated for maintaining the desired level of quality.



During the development of support for a new ODD, there will be significant information interchange between the ODD scenarios definition and the functionality implementation. This work is expected to follow an iterative approach to development, not a waterfall-style approach.

The relationship between the definition of an ODD and the implementation of support for that ODD is shown below.



(Image for illustrative purposes only; specific steps and stages are not exact.)

Verifying support for an ODD

Throughout the software development, verification via simulation-based Continuous Integration will form the backbone of our verification strategy.

To achieve this, we require that ODDs be defined in a machine-readable format. This applies both to universal parameters and to the scenario specifications. Machine-readable formats for scenarios already exist, such as OpenScenario, and it is expected that one such format will be chosen and reused.

In addition to the above, we will also pursue standard software engineering best practices for ensuring software quality, such as a minimum coverage level for unit tests (at least 85%), design and implementation reviews, use of static analysis, and black-box interface tests.

Autoware Core and Autoware Universe have different purposes and therefore have different required quality levels.

- The quality level required by any package in Autoware Core is Level 1 according to REP 2004.
- The quality level required by any package in Autoware Universe is Level 3 according to REP 2004.

Demonstrations

Each time Autoware achieves capability in a new ODD, we prefer to provide a demonstration of this capability. The demonstration serves multiple purposes.

- It can be executed on a real car in a real-world environment for real-world validation.
- It can be executed on a real car in a real-world environment for PR purposes. (Demonstration to the press, demonstration at conferences, etc.).
- It can be executed in a simulator by new and existing users to view the new functionality even without a real car or the necessary real-world environment.

When we begin developing the functionality for a new ODD, we must set the demonstration that will be used for that ODD. This demonstration will drive the *application-level requirements* for the work for that ODD. In other words, it will provide the use cases that drive requirements on Autoware that cannot be derived from scenarios. Such requirements include, for example, what external interfaces Autoware should have for communication outside of the autonomous driving stack.

Use of working groups

ODD working group

Purpose of the working group

This working group exists to define each ODD chosen by the TSC and populate the list of scenarios that must be drivable by Autoware for that ODD.

Tasks

- Define the ODD parameters universal to all scenarios.
- Define the list of required scenarios.
- Obtain approval of the ODD definition and list of scenarios from the TSC.
- Specify each scenario that does not yet exist in a machine-readable format.
- Produce visual guides to the ODD and scenarios to aide in developer understanding.

Autonomy Software working group

Purpose of the working group

This working group exists to manage the development of the Autoware software. It sets specific development milestones targeted towards the currently in-progress ODD technology target and manages the contributions of developers.

Tasks

- Manage the maintenance of the Autoware software.
- Hold regular meetings of contributors to discuss progress and blocking issues.
- Manage the development of features necessary to achieve the currently in-progress ODD technology target.