Traceable Robotic Handling of Sterile Medical Products

Verification Plan

Deliverable 1.3

Deliverable Title	D1.3 Verification Plan
Deliverable Lead:	INVITE GmbH
Related Work Package:	WP1: Requirements management and verification preparation
Related Task(s):	T1.4: verification plan
Author(s):	T. Cichon, C.-H. Coulon
Dissemination Level:	Public
Due Submission Date:	28/02/2022
Actual Submission:	28/02/2022
Project Number	101017089
Instrument:	Research and innovation action
Start Date of Project:	01.01.2021
Duration:	51 months
Abstract	In this deliverable we will first present the verification concept of TraceBot including utilizing all Digital Twin capabilities as well as the interactive robotic process channels. Afterwards, we bring this concept to a concrete verification process in terms of the acceptance test (including a check on “works as planned” and a seamless audit trail) and regarding specific discrepancies to challenge the test and thus the TraceBot system.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089), 2021-2025

Versioning and Contribution History

Version	Date	Modified by	Modification reason
v.01	12.01.2022	T. Cichon (INV)	Initial setup of document
v.02	24.01.2022	T. Cichon (INV)	Update expected content after project meeting
v.03	03.02.2022	T. Cichon (INV)	Update content after discussion with Anthony Remazeilles
v.04	11.02.2022	T. Cichon (INV)	Update for internal revision
V.1.0	11.02.2022	T. Cichon (INV)	Ready for internal revision
v. 1.1	22.02.2022	T. Cichon (INV)	Update after internal revision
V.2.0	24.02.2022	A. Remazeilles (TECN)	Revised version ready for submission

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Table of Contents

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

1 Executive Summary

Closely related to T1.4 this deliverable will describe the verification and validation plan.

T1.4 Verification plan (M4-14 M25-45; INV; CEA, AST, TUW, TECN, UOB)
This will form the basis of the acceptance tests for the demonstrator and associated evidence, to be performed in WP6 (T6.3). The target audiences include stakeholders such as members of the Advisory Board, external partners, customer, and regulatory representatives (product provider, service provider, regulation auditor). A verification matrix will check that each test matches each requirement (D1.1). Forced failures will be used to challenge the systems to prove traceability. The failure mode FMEA (D1.2) shows that each risk is under control and the verification steps to go through will be further analyzed to fit GMP (good manufacturing practice) regulations. This will provide input for verification in customer settings, although outside of the project scope. The plan will be reviewed by an internal quality auditor to demonstrate the potential for regulatory compliance.

Verification in the chemical/pharmaceutical industry is often done by a human which is trained and additionally has an "instinct" regarding unexpected deviations of a process. Bringing this combination of rule-based verification and self-awareness to the robot is the goal of TraceBot. We will approach this technically with additional *verification skills* as well as a central *Tracer* module that logs everything (each executed skill including time stamp, error messages etc.). This logging is the central building block of the validation and the final audit trail of TraceBot.

With such a system in place we then want to verify its proper execution therefore we aim to implement an *acceptance test* for our system. This acceptance test is the implementation of the *verification concept* and will be used throughout the whole project and with all developed demonstrators to verify the traceable robot actions, confirm the good integration, and provide guidance for possible improvements.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

2 Introduction

As traceability is one of the core building blocks of TraceBot the verification and validation plan is an essential part of the project. Thus, in the following we will first describe how we conceptually approach the TraceBot-verification, starting from the state of the art in human and roboticized verification procedures. After describing the verification process to be carried out by the TraceBot robot in more detail (also from a technical/implementational point of view) we aim at a concrete verification process resulting in acceptance tests descriptions. This should then be a blueprint on how to challenge and evaluate the TraceBot system at each integration milestones, and at the end of the project.

The Verification Plan will be developed first **conceptually** and then **technically**. This means in section 3 we will describe the verification concept that is then put into a more practical verification process in section 4.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

3 Verification concept

The verification concept is a three-step approach to come up with what and how we target verification in TraceBot.

We start with the three levels of verification delivered by a human operator currently performing sterility tests (section 3.1). Afterwards, we take a look at standard kinds of verification within automated systems (section 3.2) to finally present the new kinds and levels of verification targeted by TraceBot (section 3.3).

3.1 Verification by human

A lot of processes in the chemical pharmaceutical industries involve manual steps and particularly the final verification of a successful step is often done by a human.

This verification by a human can again be grouped into three different layers, exemplarily shown with the example of opening the sterility kit (cf. Fig. 1, Fig. 2, Fig. 3, Fig. 4 as well as D1.6 (Cichon & Coulon, 2022)).

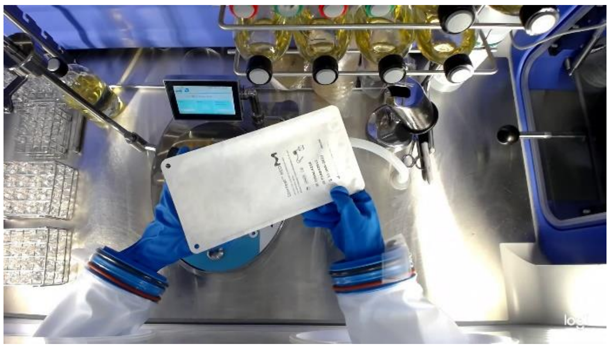
Fig. 1 take closed sterility kit

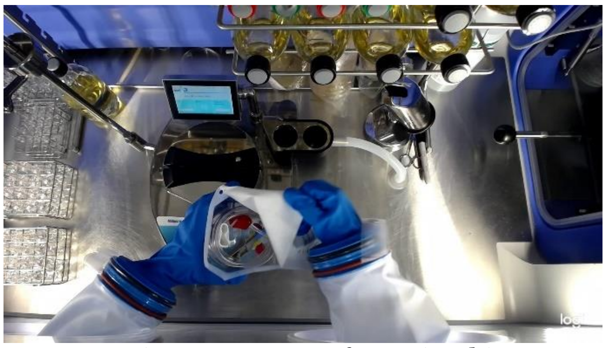
Fig. 2: process of opening kit

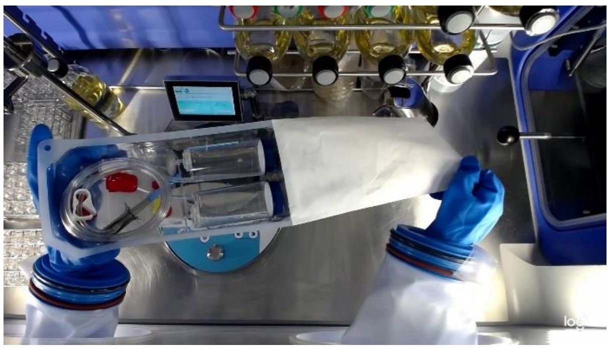
Fig. 3: finishing opening kit

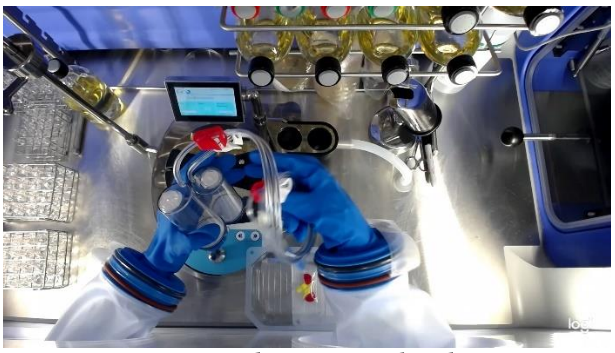
Fig. 4: take out sterility kit

First, during the process the operator performs the checks he was trained for, e.g., the operator checks that the foil is not broken before opening. This, first level verification can be seen as a **rule-based verification.**

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Secondly, as a "human build-in function", the operator recognizes the success of an action, e.g., he feels that he has gripped the foil and has not lost it in the process. As a consequence, the operator in the end can sign that the process was "executed as expected" in a sequence of successful steps. This is an automatic and human built-in audit trail and can be rephrased as **self-awareness** (or the awareness of success of own actions).

The highest level and most complex level of human verification is that a human has a generic understanding of the process so that he/she can detect every "unexpected" deviation of the "normality", e.g., the kit foil is damaged, which means the sterility of the kit may be compromised. This **generic understanding** is no longer strictly based on rules but more on the generic semantic world knowledge and experience of the human.

3.2 Verification by automation (state of the art)

After describing the human verification in the prior section, we now take a look at how state of the art automation and its verification are.

In general, based on a user-requirement specification a risk analysis is performed, which also includes an analysis such as a failure mode and effects analysis (FMEA).

For all recognized risks counter measures are defined, either organizational (e.g. the human has to document the integrity of the foil) or technically (e.g. a pressure-based measuring of the integrity of the foil)

Within the qualification all technically established counter measures are tested/ audited and are used as a basis of the following validation of the process, which also takes the organizational countermeasures into account.

This means that the standard verification requires executing technical countermeasures defined by a risk analysis and be a rule-based verification. Rule-based verification is totally aligned with the capabilities of standard robotic systems, as long as an engineer is able to implement a specific component to enable the rule verification (i.e., vision-based or tactile-based verification of the foil integrity before opening the sterility testing kit).

A negative aspect is that in a non-completely controlled environment (i.e., object location may change), the number of possible risks is very large, and the definition of countermeasure or verification means for each of them may be complex. Also, the lack of understanding or awareness of the system does not able the robot to detect unexpected situations, as he can only verify what it has been explicitly programmed to. To alleviate this, we are proposing to equip the TraceBot robot with **self-awareness** capabilities.

3.3 Verification by TraceBot (beyond state of the art)

Aligning with TraceBot's objective O1 and the aforementioned goal of adding self-awareness to robotics TraceBot will target these in a two-step approach:

First, we will allow an explicit management of technical verifications, which can also be seen as hard-coded countermeasures to verify each pre-defined step.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Secondly, we will enable the self-awareness of the robotic system by using the digital twin in-the-loop, tracing the success of all steps. Combining different perception technologies (tactile sensors, encoders, cameras etc.) with the digital twin, checking the outcome of activities, looking for specific kind of deviations will lead to an iterative test after each step between real and digital twin environments, e.g., in pre-defined features like pose of object, status of gripper, or similar.

Thus, we will have **two types of verification channels:**

1. The **Digital Twin channel**, where all robot (perception and manipulation) actions are checked by comparing them to the Digital Twin's expectation. This enables a kind of the "self-awareness" of the robot about its actions and if they are executed as expected.
2. The **Process interaction channel**, where (additional) robot actions are executed to create "functional", "visual" or "tactile" data feedback with the aim to verify if the desired robot action has been executed correctly.

Both of these channels eventually feed into the tracer and this enables the creation of the audit trail on top of the simulated expectation.

Without entering into details (see D4.1 (Vincze, Weibel, Mania, & Vial, 2022) for more information), one could say, from an implementation standpoint, that the verification takes place in additional verification skill (which accompanies each concrete skill) and everything is additionally logged (or traced) in a standalone component for now called "Tracer" which will deliver the audit-trail containing all the documentation about the operations conducted by the robotic system.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

4 Verification Process

The verification process, in contrast to the aforementioned concept, is about controlling and/or monitoring the verification result. In addition, the purpose of the verification process defined here is to provide a set of tests to be done after the integration to confirm that the system works as expected, with the expected robustness. Thus, the result should be something like a checklist that we can use as soon as a demonstrator is ready to check the progress in terms of verification.

This involves answering to the following questions:

1. What can possibly go wrong? Including the automation but especially the process itself.
2. How will the robot behave if we (en)force an error from point 1.?

In contrast to the technical implementation (WP4 and WP5) on how to detect if an action goes right or wrong here, we introduce **acceptance tests,** that are used to test if the process is done as expected.

As the complete sterility testing use case, described in D1.6 (Cichon & Coulon, 2022) is significantly long, and the robotic system will not be able to handle it completely yet from the first integration phase, we will progressively implement sub use cases. The acceptance tests should be applied in the sub use cases after each major integration in the TraceBot timeline, respectively after each milestone and should be mainly done within WP6.

Therefore, we took the decision of not writing an exhaustive verification process with all possible acceptance tests. Instead, we address the problem in a generic way. Then, in WP6, the generic acceptance tests will be specialized depending on the capabilities of the integrated system and detailed within the related validation document.

4.1 Acceptance Test

The acceptance tests of TraceBot will focus on:

1. verifying the robot **works as planned**, providing the expected capabilities, and performing the requested operations (like insert the canister into the tray).
2. Verifying a **seamless audit trail** and thus the validation capabilities

Besides these two major items, it is worth commenting that with the successive iterations of the system development we aim at a system that is able to detect any deviation in the process and furthermore, the robot should be able to recover from some of these failures (i.e a bottle grasping failed, so that the robot tries the grasp a second time).

Note that during this verification process, the audit trail content will also be verified in terms of quality of the information contained, timestamp coherence, and other relevant features that still need to be defined.

From the regulatory point of view, the detection and documentation of deviations is way more important than a complete execution of the task at hand. Therefore, major expectations for addressing point 2 are that the system **detects** that "something" is wrong, **acts** accordingly (either *stop* or go into *recovery*) and also **documents** everything.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

Detailing *how* the system detects if an action goes right or wrong is not part of WP1 but will be covered within the work packages of the technical implementation.

Example: simplified use case “fit canister to tray”.
If we set the system in good starting configuration, and we run the use case 10 times:

- Does it perform robustly the use case (time, %success)?
- Is the audit trail well documented?

When failure occurs,

- When does it take place?
- Is the error detected by the system?

Is the system logging the error? Is the system able to recover on its own?

4.1.1 Key Performance Indicators

To define which key performance indicator should be used, we focus on the application. In the chemical/pharmaceutical industry normally the main incentive is to find out if there was an issue and if this issue was documented. In the first iteration it is not about the success of the task itself and thus a “stop” of the process is not a problem, from the regulation point of view. Of course, a stop of a process can be seen as a drawback from a business point of view due to the delays it may introduce in the process if a human has to intervene.

Thus, for our acceptance test this leads to the main criteria that are

1. Logged [yes/no]
2. Overall success [yes/no]

Additional success criteria are:

3. Number of trials [N]
4. Success of Process(step) [%]
5. Time [s]
6. Number of errors [N]
7. Error detected [yes/no]
8. System stop [yes/no]
9. System recovery [yes/no]

Additional criteria can be added regarding the digital twin

10. DT used [N]
11. DT based mitigation [yes/no]
12. DT successfully used [%]

This first set of Performance Indicators may be updated and/ or additionally defined throughout the project.

4.1.2 Deviations from the manual execution

Deviations related to the conditions of execution of the tasks is another focus point as the use cases executed by the robotic system might differ from the way it is usually conducted in the human process.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

These deviations may be related to capabilities of the robotic system. For instance, the robot could not have yet the capabilities to execute a task like a human does (for instance the two canisters are connected by tubes, and the first experimentation will be done using a unique robotic arm). Also, some adjustments may be proposed as it may not make sense to replicate exactly the human operations with the robot (e.g., labeling something with a pen).

The analysis of these deviations is important to highlight possible lines of actions for the following integration stage, and /or to identify possible adjustments of the process to be more robot friendly, which will be studied within T1.3 (Robot-friendly design guidebook).

4.2 Challenges for the system

The acceptance test previously described is focused on looking at the capability of the robot to perform the robotic operations it is designed for, documenting the execution, detecting any deviations and acting in a feasible way. These three steps are the central goals of the verification and validation plan.

On top of that, to **actively challenge the system** we want to define challenges that should prove a validated task execution. As the verification layer should detect any abnormal situations, the possibilities are quite numerous (anything but the normal situation). To make it more manageable, we aim at developing a set of specific challenges for the robotic system which we would expect it to be able to cope with.

Thus, we aim at developing a set of **categories** for challenges, like the following, based on the use case description in D1.6 (Cichon & Coulon, 2022):

a) Sensing,
b) Grasping,
c) Manipulating,
d) Digital Twin understanding

Indeed, challenging situations could occur in the technical realization of vision-related tasks, like the identification or pose-estimation of objects in the scene, or in manipulation tasks, like grabbing an object or inserting one object into another. Of course, there might be some overlaps regarding vision-based manipulation etc. but still the main categories could clearly be identified.

Another field of challenges could be introduced with the Digital Twin and the NEEM framework (see WP4 and WP5) where for example mismatches in the ontology and/or simulation of the Digital Twin could result in wrong reasonings.

Additionally, the categories of challenges can also be sub-divided into different complexity levels:

- Sensing
  - Vision:
    - Expected object not present in scene
    - Object partially occluded
    - Expected object present in several instances
  - Tactile:
    - Object not present in the gripper as expected

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

        - Object not at expected position in hand
        - Other object present in the hand
        - Heavier object in hand (to provoke slippage)
        - Gripper intentionally opened
- Grasping:
    - Gripper badly positioned with respect to object
    - Object not present
- Manipulation:
    - Object badly positioned in hand
    - Target location slightly displaced
    - Obstacle in the path to the target
- Digital twin understanding:
    - Obstacle placed in the scene
    - Target location not accessible
    - Object displaced in scene

These different categories will be further on detailed and completed according to the concrete use cases that are implemented and ready for evaluation.

These challenges may occur during the normal execution[1] (as described before) but they will also artificially be introduced by the integrator to challenge the system and confirm the robotic system verification capability.

Example: Simplified Use Case "insert canister into tray"

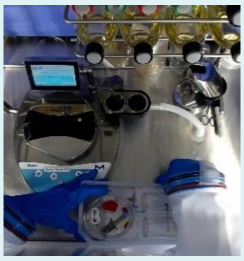
Fig. 5: Object localization

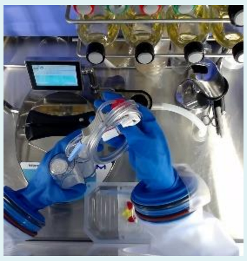
Fig. 6: Grab canister

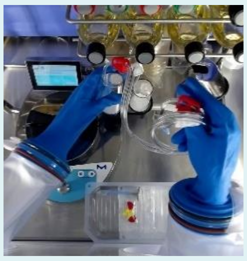
Fig. 7: Move to Tray

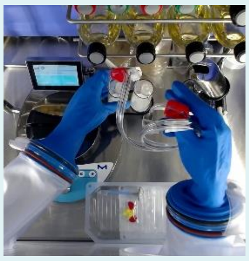
Fig. 8: Release canister

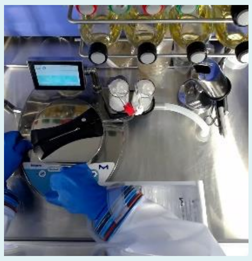
Fig. 9 Inspection of canisters

Looking again at the simplified use case, (cf. Fig. 5 - Fig. 9), we can define perturbations meaning:

- Object localization: launch without the object being present

[1] This statement needs to be confirmed as theoretically in the chemical pharmaceutical industry deviations in the setup or execution are not tolerated and as our system is working independently based on a given starting point there should be no challenges at all. Thus we have to evaluate if the mentioned challenges are feasible challenges that will occur during normal operation.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

- Grab object: take out the object just before the grasping start, move of few mm the object before the grasp
- Insert canister: slightly move the tray or the object in the gripper hand
- Inspection: After the insertion, a visual verification is used to confirm the good insertion. We can take out the object, place it badly,

Each of these challenges are then to be evaluated within the process execution with regard to

- Logged [yes/no]
- Overall success [yes/no]
- Additional success criteria, especially
  - Error detected [yes/no]
  - System stop [yes/no]
  - System recovery [yes/no]

This will then result in a checklist that can be used in the evaluation of each demonstrator and could look like Table 1.

Table 1 Exemplary filled Checklist for one run (out of many)

Run: #1/10	UseCase: Simplified Use Case "Insert Canister"									
	Main criteria	Logged [yes/no]	Overall success [yes/no]	Additional success criteria	Error detected [yes/no]	System stop [yes/no]	System recovery [yes/no]	…	…	Time [s]
Object localization:		✓	✓		✓	✓	✗			30
Grab object		✓	✗		✓	✓	✗			12
Insert canister		✓	✓		✓	✗	✓			51
Inspection		✓	✓		✗	✗	✗			41
Overall time [s]	134									

Such a checklist is still under development and will be updated continuously especially for being used in all integration activities. In the end, this checklist needs to be compared with the output of the Tracer through the audit trail.

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)

5 Deviations from the workplan

none

6 Conclusion

The verification process we envision is a combination of standard hard-coded countermeasures and a continuous tracing and verification involving real data and digital twin data, which is one of the main targets of TraceBot.

With the acceptance test proposed we are aiming at a concrete checklist for challenging the system to derive its verified execution of process-steps. This generation of this checklist is an iterative approach and will be done accordingly to the technical capabilities of the system, but the overall main success criteria will always stay as the success itself and the traceability will always be of paramount importance.

The example stated throughout this deliverable is the first of many “simplified use cases” and as for now the "insert canister” is used to point out the major verification and validation steps further sub-use-cases like “insert needles” or others will be used for refining also the verification and validation plan. This can then iteratively be used in the demonstrator from WP6 to cross-check the challenges and system responses proposed in this deliverable.

This generic validation process will be the framework of the verification process conducted in WP6 at each milestone. In that context, the generic model will be specialized to the specific use case evaluated.

According to the Gantt chart, this verification plan will be completed after the next milestone, to complete the testing items, and, possibly, to define a set of concrete benchmarking operations which could be used aside the concrete tests mentioned here to compare the capabilities of robotic systems to conduct the operations required in such pharmaceutical application.

7 References

Cichon, T., & Coulon, C.-H. (2022). *TraceBot D1.6 Use Case Specification.*

Vincze, M., Weibel, J., Mania, P., & Vial, F. (2022). *TraceBot D4.1 Traceability framework for laboratory automation.*

Horizon 2020

TraceBot receives funding from the European Union's H2020-EU.2.1.1. INDUSTRIAL LEADERSHIP programme (grant agreement No 101017089)