Setting up a digital production environment

10 January 2025, 13:39

Filiep Vincent

Introducing AMR/AGV in a production environment - Part 2

Logistics operations add little value, but they are necessary to keep production on track. After all, materials, components and tools need to be in the right place at the right time. In this four-part blog series, we want to share our experiences concerning the role autonomous vehicles can play here and how to implement such an approach.

In this second part, we reflect on the digital environment that flexibly controls autonomous vehicles and, by extension, all operations.

Digitisation of production environment with AGVs

From unified namespace (UNS)...

A **unified namespace (UNS)** is a coherent and consistent system that names entities within a given environment or context . Such a system means that each entity – whether a file, a device, a user or some other source – is identified uniquely. It also ensures that all entities are easily accessible.

One of the main benefits of a UNS is improved **usability and accessibility**. A standardised naming convention makes it easy for users to find and refer to data sources without ambiguity or confusion.

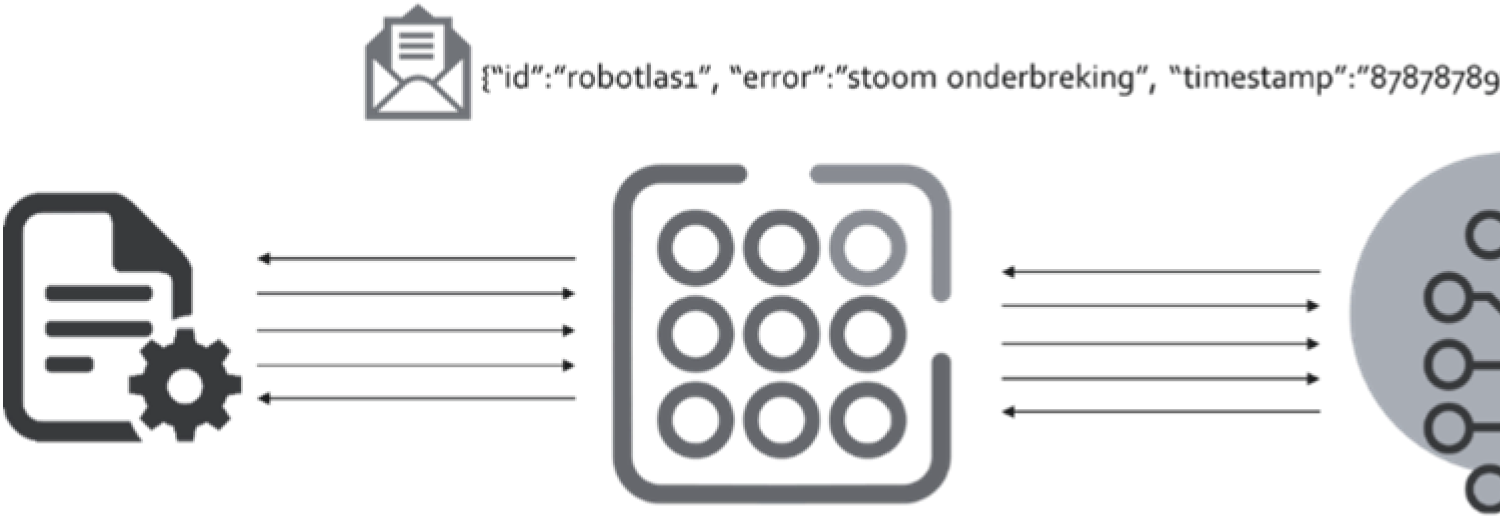
In addition, a UNS facilitates **interoperability and integration** between different systems and platforms. When multiple systems use the same namespace, it becomes easier to exchange data, communicate between systems and perform cross-platform operations. This interoperability is crucial to setting up an ‘industry 4.0’ production environment, in which data needs to be freely available to all systems (ERP, MES, WMS etc.).

A UNS also improves **scalability and flexibility**. As the system grows and evolves, it integrates new entities into the existing namespace seamlessly, without disrupting workflows or requiring significant changes. This scalability means that the namespace remains efficient and manageable, even as it grows larger and more complex.

...to message queuing telemetry transport (MQTT)

Message queuing telemetry transport (MQTT) is a lightweight **publish-subscribe messaging protocol**. Sensors, machines, devices and other sources of data publish messages on topics. In doing so, they use the naming system of the UNS model and serve as communication channels. ERP and MES software systems, for example, subscribe to these topics in order to receive the messages.

An MQTT broker can be compared to a large cabinet with **small mailboxes**. Thanks to UNS, each mailbox has a specific data source with a unique name and contains a structured message. A system or data source can put (publish) a message in one of these mailboxes; other systems subscribe and thus receive the published message. Although only one unique publisher or data source is possible per mailbox, multiple systems can subscribe to the same mailbox. This makes it possible to share the same data with different, even competing, systems. The decoupling of publishers and subscribers makes scalable and flexible communication architectures conceivable, as the number of mailboxes that can be created is limitless.


{"id":"robotlas1", "error":"stoom onderbreking", "timestamp":"87878789

Sirris

MQTT supports **quality of service (QoS)** levels, allowing users to strike a balance between the guarantee that messages will be delivered and the network overhead. In addition, there is support for **last will and testament (LWT)** messages, which the MQTT broker sends when a client unexpectedly disconnects. This feature ensures that disconnected clients are handled smoothly and makes robust communication architectures possible.

Overall, the **simplicity, efficiency and reliability** of MQTT make it a popular choice for a wide range of IoT (Internet of things), M2M (machine-to-machine) and mobile applications.

Concrete example

Within the 4.0 made real demonstrator, the namespace is based on the physical location of the workstation within Sirris.

Business	Location	Zone	Workstation	UNS
Sirris	Genk	warehouse	reception	Sirris/Genk/warehouse/reception
Sirris	Genk	warehouse	reception	Sirris/Genk/warehouse/reception

Sirris	Genk	warehouse	reception	Sirris/Genk/warehouse/inventory
Sirris	Genk	shopfloor	prep_zone	Sirris/Genk/shopfloor/prep_zone
Sirris	Genk	shopfloor	ko-ga-me	Sirris/Genk/shopfloor/ko-ga-me
Sirris	Genk	shopfloor	ko-ga-me	Sirris/Genk/shopfloor/haas_cnc

The structure of the message that the MQTT broker sends is fixed for the different data sources. The structure of the warehouse that publishes on the UNS topic ‘Sirris/Genk/warehouse/inventory’ is as follows:

```
{"timestamp":"YYYY-mm-dd hh:mm:ss",
"productID": "R1000",
"locationID": "OR1-01-01",
"quantity": "100",
"unit": "pieces"/"boxes"/"gram"
}
```

The message includes the following:

- a timestamp;
- the reference of the product that is brought into or out of the warehouse (productID);
- the exact location within the warehouse where the action takes place (locationID);
- the quantity;
- the associated unit that goes into or out of the warehouse.

The namespace thus describes the **exact action.** After publication, the WMS reads the message and processes it to adjust the stock. The MES does this to indicate the start of an order

This article is part of a four-part series on introducing AMR/AGV within a manufacturing environment. Would you like to read more? Discover our

other articles in this series:

Part one | Choosing the most suitable production strategy >

Part three | The fleet manager within the demonstrator >

Part four | Everything you need to know before starting an AGVsim project >

Authors

Filiep Vincent