

Het Fraunhofer Project Centre op de Universiteit Twente is op dit moment bezig met het ontwikkelen van een testbed. Dit testbed is een omgeving waarin bedrijven hun productielijnen kunnen testen, maar daarvoor niet per se deze lijnen stil te hoeven leggen. Dit kan omdat het testbed bestaat uit een synthetische omgeving waarin digital twins van machines gebruikt worden om hier scenario's mee te kunnen simuleren die real time worden aangepast. Het geen wat het testbed anders maakt dan normaal is dat dit een testomgeving moet worden die meerdere scenario's moet kunnen uitvoeren van verschillende bedrijven en verschillende stakeholders. Dit maakt dat er in plaats van een vast geprogrammeerde uitvoering, er een gedrag ontwikkeld moet worden dat alle (onbekende) scenario's en verstoringen moet kunnen reageren. Daarom is de hoofdvraag van dit onderzoek dan ook: "Hoe kan je het gedrag van het testbed het best modelleren?"

Om dit uit te kunnen voeren is eerst gekeken naar wat de doelen en eisen voor het testbed zijn, om daarna te concluderen welke eisen er dan aan het functionele gedrag van het testbed zitten. De eisen zijn dat de functionaliteit modulair, schaalbaar, slim/autonoom, aanpasbaar en toegankelijk moet zijn op elke laag van het systeem.

Voor het opbouwen van het systeem dat functioneel aan de eisen voldoet is getracht een bouwblok te maken waarmee alle functies in een scenario omschreven kunnen worden. Op deze manier kan de modulariteit en schaalbaarheid verkregen worden. Dit blok is opgezet volgens het IDEF0 model, waar de in- en uitgaande pijlen gecategoriseerd zijn volgens productie aspecten en functies uit verschillende lagen bestaan. Dit blok is gevalideerd door het te toetsen ten opzichte van meerdere scenario's, waarvan één scenario de uiteindelijke demonstrator gaat uitvoeren, door het model hierop toe te passen. De hoogste laag van een functie kan worden gezien als een eenheid met één van de 6 domeinen binnen de productie: productie, assemblage, inspectie, transport, opslag en onderhoud. De lagen daaronder beschrijven in steeds gedetailleerderestappen het process van de bovengelige eenheid.

Na het ontwikkelen en valideren van het bouwblok is er geanalyseerd hoe het autonome gedrag, door middel van het toepassen van dit model, ingebouwd kan worden. Dit heeft te maken met het bepalen van de flexibiliteit per functie en de manier waarop het systeem keuzes maakt. De flexibiliteit per functie is te bepalen aan de mate waarin instructies worden doorgevoerd over alle lagen van het model. Voor een meer flexibele functie ligt de instructie meer open, en wordt deze op het hoogste niveau gegeven waarna het mechanisme zelf moet uitzoeken hoe het daar komt. Ook kan het systeem op twee manieren keuzes maken, namelijk intra-functioneel, binnen het uitvoeren van een functie en inter-functioneel, namelijk het kiezen tussen welke functies uitgevoerd worden. De uiteindelijke keuzes die gemaakt worden zijn afhankelijk van de strategie die getest wordt. Hier zit ook een zelf-lerend aspect in, namelijk dat in het geval een afwijking waar het systeem niet intra- of inter-functioneel kan oplossen, dit voorgelegd wordt aan de gebruiker die vervolgens een keuze maakt. Dit wordt dan onthouden en kan in vergelijkbare manieren gebruikt worden.

Als alles samengevoegd wordt, kan er door middel van het ontworpen model een functioneel gedrag in kaart gebracht worden, door gebruik te maken van het bouwblok. Voordat het systeem gaat werken, moeten er nog wel keuzes worden gemaakt in strategie of prioriteiten, gezien het systeem keuzes maakt afhankelijk hiervan.

The Fraunhofer Project Centre at the University of Twente is developing a testbed, which is a pilot production environment where different companies and stakeholders can test their production environments without hampering with their primary process. This is possible because the testbed is a synthetic environment, where digital twins of assets are used to run different scenarios that are connected with real time data. The unique aspect of the testbed is that it must be suitable to test different scenarios and thus be reconfigurable. So instead of programming a fixed order of operations, there is a need for a general behaviour which can handle all (unknown) scenarios. Therefore, the main research question of this project is: "How can the behaviour of the testbed be modelled?"

First, the goals and requirements of the testbed were assessed, after which the requirements for the functionality of the testbed could be determined. The requirements for the functionality are that it is modular, scalable, autonomous (or 'smart'), and accessible on all different levels of the system.

To build up this functional model, it is tried to obtain a general building block that can describe all functions in a scenario. In this way, the modularity and scalability are tried to be obtained. This block is set up according to the IDEF0 model, where the in- and outgoing arrows are categorised with manufacturing aspects and functions can be described with levels. This block was validated by applying it to several different scenarios, one of which is the scenario developed for the demonstrator. The highest level of a function can be seen as one of the 6 identified manufacturing domains being production, assembly, inspection, storage, transport and maintenance. The lower levels are more detailed functions of the upper function.

After the building block was developed and validated, the method of building in of the autonomous behaviour by using the functional model was analysed. This has two aspects that need to be determined, the flexibility for each function and the decision making process. The flexibility for each function can be described by the way the instructions are given. For a more flexible function, the instructions are more 'open' and given as a responsibility on the highest level of the function, whereas for a less flexible function, the instructions are set for all levels of a function.

The system can make decisions in two ways: intra-functional decision making, which is decision making within the execution of one function, and inter-functional decision making, where it is decided upon what function is executed. The decision making is dependent on the strategy that is tested in the scenario. This also has a self-learning aspect, because if there is a derivation where the system cannot use its own decision making the user will be informed and asked to make a decision. This decision can be saved and used in similar events.

If everything is put together, the developed model can be used to describe the behaviour of the testbed, by making use of the building blocks provided. Before the system will start working however, a choice of a strategy or priorities will have to be made, as the system will make decisions based on these.