



## Data Article

## FASTory digital twin data

Wael M. Mohammed\*, Jose L. Martinez Lastra

*FAST-Lab, Faculty of Engineering and Natural Sciences, Tampere University, P.O. Box 600, Tampere FI-33014, Finland*

## ARTICLE INFO

*Article history:*

Received 22 January 2021

Revised 17 February 2021

Accepted 25 February 2021

Available online 26 February 2021

*Keywords:*

Digital twin

Data engineering

Linked data

Discrete manufacturing process

Assembly process

## ABSTRACT

The vast adoption of machine learning techniques in developing smart solutions increases the need of training and testing data. This data can be either collected from physical systems or created using simulation tools. In this regard, this paper presents a set of data collected using a digital twin known as the FASTory Simulator. The data contains more than 100 K events which are collected during a simulated assembly process. The FASTory simulator is a replica of a real assembly line with web-based industrial controllers. The data have been collected using specific-developed orchestrator. During the simulated process, the orchestrator was able to record all the events that occurred in the system. The provided data contains raw JavaScript Object Notation (JSON) formatted data and filtered Comma Separated Values (CSV) formatted data. This data can be exploited in machine learning for modelling the behaviour of the production systems or as testing data for optimization solution for the production system. Finally, this data has been utilized in a research for comparing different data analysis approaches including Knowledge-based systems and data-based systems.

© 2021 The Author(s). Published by Elsevier Inc.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

\* Corresponding author.

E-mail address: [wael.mohammed@tuni.fi](mailto:wael.mohammed@tuni.fi) (W.M. Mohammed).

**Specifications Table**

Subject	Industrial Engineering, Manufacturing Engineering
Specific subject area	Discrete manufacturing data includes the events of simulated assembly process
Type of data	CSV, JSON
How data were acquired	The data has been collected using a digital twin known as the FASTory Simulator [1]. This simulator mimics an actual assembly system located in Tampere university. The simulator was controlled by an inhouse developed orchestrator for the purpose of data collection using web services.
Data format	Raw and filtered
Parameters for data collection	The collected data contains 106,154 events during a simulated assembly process [2]. The collection process lasted for 12 h distributed on 3 days. For the data collection, an orchestrator has been set for managing 3 pallets at the same time with continues utilization.
Description of data collection	The data was collected using an orchestrator which acted as event logger for this experiment. The orchestrator uses REST-formatted HTTP notifications for generating the events. The orchestrator needed to subscribe to all notifications in order to collect the data. Besides, the orchestrator managed the assembly of 1017 simulated product. [3]
Data source location	FASTory Simulator [3] Future Automation Systems and Technologies Laboratory (FAST-Lab) Tampere University Tampere, Finland
Data accessibility	Dataset: <a href="https://doi.org/10.23729/7e299722-c246-4695-990f-dbe5a7839ad2">https://doi.org/10.23729/7e299722-c246-4695-990f-dbe5a7839ad2</a> [2] Software (simulator and orchestrator): <a href="https://doi.org/10.5281/zenodo.4541008">https://doi.org/10.5281/zenodo.4541008</a> [3]
Related research article	R.F. Borja, M. Wael M., A. Mussawar, I. Sergii, Z. Jiayi, H. Robert, M.L. Jose L., Comparing ontologies and databases: a critical review of lifecycle engineering models in manufacturing, Knowledge and Information Systems, In Press.

**Value of the Data**

- The collected data include more than 100 K events generated during a simulated assembly process of electronic devices using the FASTory Simulator. The data can be important for the purposes of studying different techniques and methods for balancing and optimizing discrete processes.
- The data can be utilized by researchers and developer of manufacturing systems that are specialized in optimizing manufacturing processes. The data can also be important for companies who develop Manufacturing Execution Systems (MES) and Enterprise Resources Planning (ERP) systems [4].
- The FASTory Digital Twin data can be exploited in modelling discrete manufacturing processes or testing and validating optimization solutions for balancing the utilization of the manufacturing resources. In addition, the data can be used for training machine learning models that targets replicating an industrial discrete manufacturing process.

**1. Data Description**

*1.1. Data generation using the FASTory digital twin*

The FASTory line is an education and research assembly line located at the FAST-Lab in Tampere University Tampere, Finland. The assembly line was utilized for assembling mobile phones during the early 2000s. Afterwards, the line was relocated to the university and retrofitted with web-based Remote Terminal Units (RTUs) to control the conveyors and the robots alongside handling other sensors and actuators as Fig. 1 shows on the right. In addition, the line is transformed to mimic the assembly process by drawing main components of mobile phones (frame, screen, and keyboard) with 3 different models and 3 possible colours for each part. These



Fig. 1. On the left, the FASTory line product and on the right, the FASTory line [1].

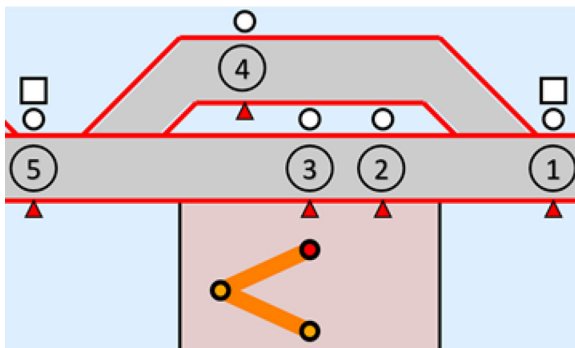


Fig. 2. The location of zones in a workstation numbered from 1 to 5.

variations generate a total of 729 different product. As show in Fig. 1 on the left, the pallets have been modified to hold the papers which represents the main component in the assembly. Once the paper is placed on a pallet, the drawing processes of the three parts are executed. Then the paper is extracted from the pallet. In this regard, the paper with the drawn phone represent a fully assembled product.

The FASTory line consists of 10 identical workstations (numbered from 2 to 6 and 8 to 12) for executing the drawing processes, one workstation (numbered as 1) for loading and unloading the papers and one workstation (numbered as 7) for loading and unloading pallets. Each workstation is equipped with RFID for recognizing the incoming pallets. In addition, each workstation (2–6 and 8–12) include a main conveyor and bypass conveyor. The main conveyor is used to locate the pallet in the position for drawing the mobile parts. Meanwhile, the bypass conveyor is used to skip the workstation without disturbing the ongoing process. For workstation 1 and 7, only main conveyor is used. Both main and bypass conveyors are divided into zones. These zones are numbered from 1 to 5 as Fig. 2 depicts. Each zone is equipped with a proximity sensor to detect the presence of the pallet and with a pneumatic actuator to hold the pallet in the zone. The functionality of these zones include:

- Zone 1 for identifying the incoming pallet;
- Zone 2 for holding 1 pallet in the workstation while the robot is executing a drawing process. This zone works as a buffer for queueing the pallet;
- Zone 3 is dedicated for executing the drawing process;

```

{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "event",
  "type": "object",
  "properties": {
    "timeStamp": {"type": "integer"},
    "event": {
      "type": "object",
      "properties": {
        "id": {"type": "string"},
        "senderId": {"type": "string"},
        "payload": {
          "type": "object",
          "properties": {
            "palletId": {"type": "string"},
            "recipe": {"type": "string"},
            "color": {"type": "string"},
          },
          "required": ["palletId"]
        }
      },
      "required": ["id", "senderId", "payload" ]
    }
  },
  "required": ["timeStamp", "event"]
}

```

Fig. 3. JSON schema of the events.

- Zone 4 is used to bypass the workstation and allows holding the pallet to avoid any collision in the conjunction of the main and the bypass conveyors. This zone does not exist in workstation 1 and workstation 7;
- Zone 5 represents the exit of the workstation. This zone is overlapping with the next workstation's zone 1.

As mentioned above, the FASTory line is controlled by web based RTUs that use the HTTP protocol. In fact, each conveyor and robot is controlled by an RTU. In total, 23 RTUs (12 conveyors and 11 robots) are used for controlling purposes. These RTUs accept HTTP services to execute operations like drawing, paper handling and pallets handling. In addition, the RTUs employ publish/subscribe mechanism for notifying third party applications about the occurring events. The schema of these events is depicted in Fig. 3. As shown in the figure, each event contains a time stamp representing the issuing time, event id identifying the event type and sender id identifying the equipment that generated the event. Afterwards, each event includes a payload that consists of the event-specific information based on the event id. Based on the event's description, the following are the possible events that can be generated by the FASTory line:

- Z1\_Changed: this event is triggered once a pallet arrives or leaves the zone. If the pallet arrives to the zone, the payload will contain the pallet id. If the pallet leaves the zone, the payload will contain "-1" as pallet id;
- Z2\_Changed: similar to Z1\_Changed but for zone 2, see Figs. 4 and 5;
- Z3\_Changed: similar to Z1\_Changed but for zone 3;
- Z4\_Changed: similar to Z1\_Changed but for zone 4;
- Z5\_Changed: similar to Z1\_Changed but for zone 5;
- DRAW\_START\_EXECUTION: this event is triggered once the process of drawing is started. The payload includes the pallet id where the drawing is conducted and the recipe of the drawing, see Fig. 7. the recipe represents the part and model of the phone;
- DRAW\_END\_EXECUTION: this event is triggered once the process of drawing is ended. The payload includes the pallet id where the drawing has been conducted, the recipe of the drawing and the colour of drawing, see Fig. 6;
- PAPER\_LOADED: is triggered once the pallet receives a new paper. The payload includes the pallet id, see Fig. 9;

```

{
  "timeStamp":1501751782579,
  "event":{
    "id":"Z2_Changed",
    "senderId":"SIM_CNV11",
    "payload":{
      "palletId":"1501749696163"
    }
  }
}

```

**Fig. 4.** Example of zone changed event once the pallet leaves the zone.

```

{
  "timeStamp":1501751789604,
  "event":{
    "id":"Z2_Changed",
    "senderId":"SIM_CNV12",
    "payload":{
      "palletId":"-1"
    }
  }
}

```

**Fig. 5.** Example of zone changed event when the pallet arrives to the zone.

```

{
  "timeStamp":1501751790656,
  "event":{
    "id":"DRAW_END_EXECUTION",
    "senderId":"SIM_ROB4",
    "payload":{
      "palletId":"1501749696165",
      "recipe":"1",
      "color":"BLUE"
    }
  }
}

```

**Fig. 6.** Example of draw end execution event.

- PAPER\_UNLOADED: is triggered once the paper is unloaded form the pallet. The payload includes the pallet id, see Fig. 8;
- PEN\_CHANGED: this event is triggered once the robot changes the colour of the pen. This event is not included in the data set due to the fixed colour for each workstation.

**Table 1**  
Description of the fields of the filtered and rearranged data.

Field name	Data type	Possible values
timeStamp	Integer	Integer represents the time according to universal time. Which is calculated since 1 January 1970 00:00:00 in milliseconds
eventId	String	Z1_Changed, Z2_Changed, Z3_Changed, Z4_Changed, Z5_Changed, PAPER_LOADED, PAPER_UNLOADED, DRAW_START_EXECUTION, DRAW_END_EXECUTION
senderId	String	SIM_CNV1, SIM_CNV2, SIM_CNV3, SIM_CNV4, SIM_CNV5, SIM_CNV6, SIM_CNV7, SIM_CNV8, SIM_CNV9, SIM_CNV10, SIM_CNV11, SIM_CNV12, SIM_ROB1, SIM_ROB2, SIM_ROB3, SIM_ROB4, SIM_ROB5, SIM_ROB6, SIM_ROB8, SIM_ROB9, SIM_ROB10, SIM_ROB11, SIM_ROB12
palletId	String	String of integer with 15 digits.
recipe	String	1,2,3,4,5,6,7,8,9 (1,2,3 for frame, 4,5,6 for screen, 7,8,9 for keyboard)
colour	String	red, green, blue

```
{
  "timeStamp":1501751786648,
  "event":{
    "id":"DRAW_START_EXECUTION",
    "senderId":"SIM_ROB4",
    "payload":{
      "palletId":"1501749696165",
      "recipe":"1"
    }
  }
}
```

**Fig. 7.** Example of draw start execution event.

```
{
  "timeStamp":1501751142646,
  "event":{
    "id":"PAPER_UNLOADED",
    "senderId":"SIM_ROB1",
    "payload":{
      "palletId":"1501749696165"
    }
  }
}
```

**Fig. 8.** Example of paper unloaded event.

1.2. Raw and filtered data

Besides sharing the raw data as JSON files, a filtering process has been conducted to rearrange the data as CSV. This data includes the similar content as described above but in computer-friendly format. Table 1 presents the description of the filtered data. It is important to mention in the CSV formatted data, the field recipe and colour are empty for the events that do not include a colour or a recipe.

```

{
  "timeStamp":1501751149235,
  "event":{
    "id":"PAPER_LOADED",
    "senderId":"SIM_ROB1",
    "payload":{
      "palletId":"1501749696165"
    }
  }
}
    
```

Fig. 9. Example of paper loaded event.

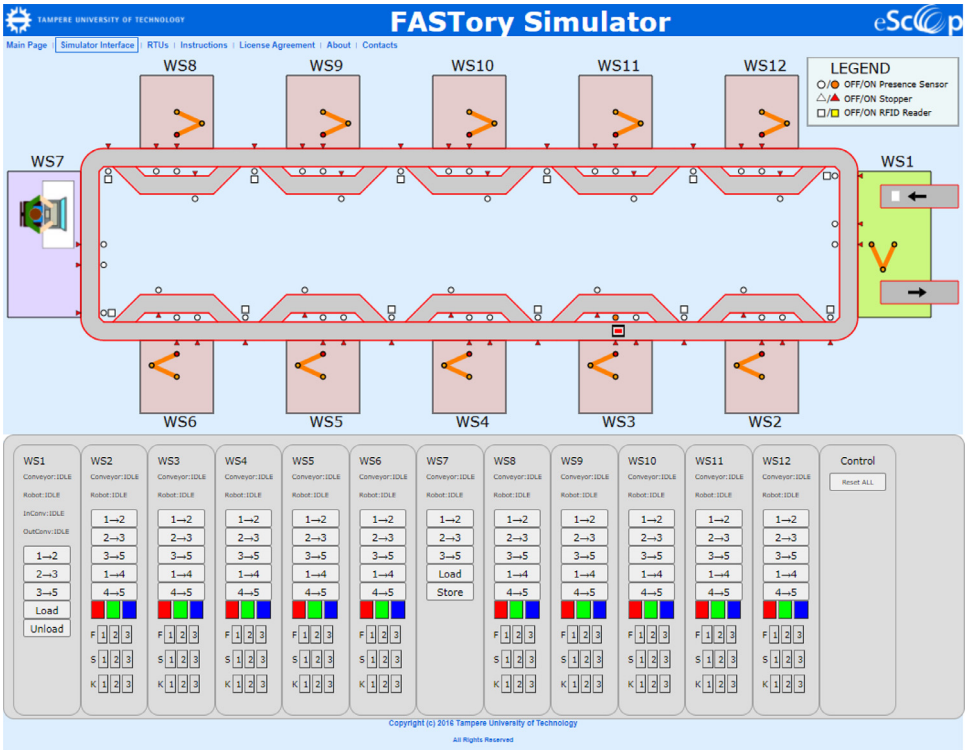


Fig. 10. FASTory digital twin [3,5].

## 2. Experimental Design, Materials and Methods

During the eScop research project [6], the FASTory Simulator<sup>1</sup> has been developed to help the project developers to test the various project applications remotely [7,8]. This allowed the developers to continuously develop and test the applications without producing any disruption or harm. For this paper, the collected data has been gathered using the FASTory simulator. This

<sup>1</sup> Accessible using: <http://130.230.141.228:3000/>.

simulator is a digital twin that mimics and simulates the services, data, timing and events. Fig. 10 depicts the interface of the simulator.

The process of gathering the data involved developing an orchestrator that subscribes to all events in the simulator and then start production process of large number of orders aiming at collecting as much data as possible. In this regard, the orchestrator starts by creating a product and then map it to the pallet. Afterwards, the orchestrator moves the pallet to load the paper, draw the 3 main part and then unload the paper. This process is repeated to finish all waiting products. For the purpose of collecting data, maximum 4 pallets were introduced at the same time during the long production process to avoid any misbehaviour or collisions.

## Ethics Statement

The authors declare that the work does not involve the use of human subject or animal experiments which need for any informed consent.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships which have or could be perceived to have influenced the work reported in this article.

## Acknowledgement

The research work by Wael M. Mohammed is supported by the Doctoral School of the President call 2018 of the Tampere University of Technology.

## Supplementary Materials

Supplementary material associated with this article can be found in the online version at doi:[10.1016/j.dib.2021.106912](https://doi.org/10.1016/j.dib.2021.106912).

## References

- [1] W.M. Mohammed, A. Lobov, B.R. Ferrer, S. Iarovy, J.L.M. Lastra, A web-based simulator for a discrete manufacturing system, in: Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), 2016, pp. 6583–6589, doi:[10.1109/IECON.2016.7793563](https://doi.org/10.1109/IECON.2016.7793563).
- [2] W.M. Mohammed, FASTory Digital Twin Dataset, (2021) 0. doi:[10.23729/7E299722-C246-4695-990F-DBE5A7839AD2](https://doi.org/10.23729/7E299722-C246-4695-990F-DBE5A7839AD2).
- [3] W.M. Mohammed, Waelut/DigitalTwin 1.0.1, Zenodo, 2021. doi:[10.5281/ZENODO.4541008](https://doi.org/10.5281/ZENODO.4541008).
- [4] W.M. Mohammed, B.R. Ferrer, S. Iarovy, E. Negri, L. Fumagalli, A. Lobov, J.L.M. Lastra, Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems, *Int. J. Comput. Integr. Manuf.* 31 (2018) 262–274, doi:[10.1080/0951192X.2017.1407874](https://doi.org/10.1080/0951192X.2017.1407874).
- [5] R.F. Borja, M. Wael M., A. Mussawar, I. Sergii, Z. Jiayi, H. Robert, M.L. Jose L., Comparing ontologies and databases: a critical review of lifecycle engineering models in manufacturing, *Knowl. Inf. Syst.* (2021).
- [6] Embedded Systems Service-based Control For Open manufacturing and Process automation | E-SCOP Project | FP7 | CORDIS | European Commission, (n.d.). <https://cordis.europa.eu/project/id/332946>. Accessed January 12, 2021.
- [7] S.O. Afolaranmi, B.R. Ferrer, W.M. Mohammed, J.L.M. Lastra, M. Ahmad, R. Harrison, Providing an access control layer to web-based applications for the industrial domain, in: Proceedings of the IEEE 15th International Conference on Industrial Informatics (INDIN), 2017, pp. 1096–1102, doi:[10.1109/INDIN.2017.8104926](https://doi.org/10.1109/INDIN.2017.8104926).
- [8] S. Iarovy, W.M. Mohammed, A. Lobov, B.R. Ferrer, J.L.M. Lastra, Cyber-physical systems for open-knowledge-driven manufacturing execution systems, in: Proceedings of the IEEE, 104, 2016, pp. 1142–1154, doi:[10.1109/JPROC.2015.2509498](https://doi.org/10.1109/JPROC.2015.2509498).