

# Autofunk: an Inference-Based Formal Model Generation Framework for [Michelin] Production Systems.

William Durand, Sébastien Salva – June 25, 2015 / FM'15





# Quick Tour @ Michelin

**A factory** is divided into several **workshops**,  
one for each step of the manufacturing process.

A **production system** is composed of devices, production machines, and one or more software to control them.

❶ In our case, we target a single workshop only.

Software exchange *information* with points and machines by sending and receiving **production messages**.

```
17-Jun-2015 23:29:59.50|17011|MSG_IN  [pid: 1] [nsec: 8] [point: 1] ...
```

```
17-Jun-2015 23:29:59.61|17021|MSG_OUT [pid: 1] [nsec: 8] [point: 3] ...
```

```
17-Jun-2015 23:29:59.70|17011|MSG_IN  [pid: 2] [nsec: 8] [point: 2] ...
```

**Production messages** are exchanged in a binary format (custom protocols), through **centralized messaging systems**.

Each production message is tied to a **product** (e.g. tire),  
identified by a **product identifier (pid)**.

Gathering all production messages related to a product  
allows to retrieve **what happened** to it.

# Background

# Development Teams POV

- 100+ applications running in **production**
- Not (fully) covered by tests
- Documentation most likely outdated
- MUST be maintained for ~20 years!

# Customers (Factories) POV

- Stability over anything else
- Maintenance periods are planned,  
but rather long (> 1 week)
- 1h (unexpected) downtime = 50k \$

**Testing** such production systems is **complex**, and takes a lot of time as it implies the physical devices, and there are **numerous behaviours**.

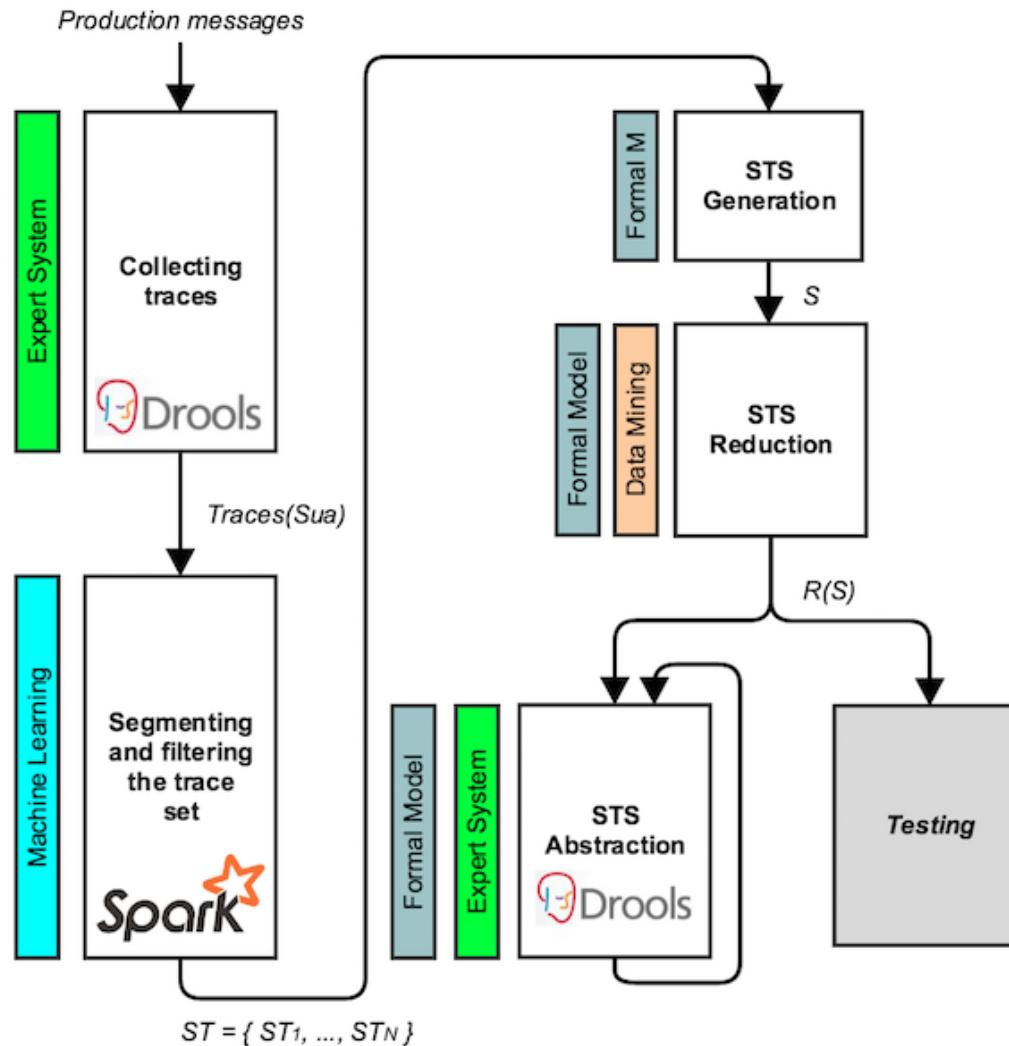
These behaviours could be formally described into a model.  
But writing such models is an **heavy task** and **error prone**.

- ❶ Not suitable for Michelin applications.

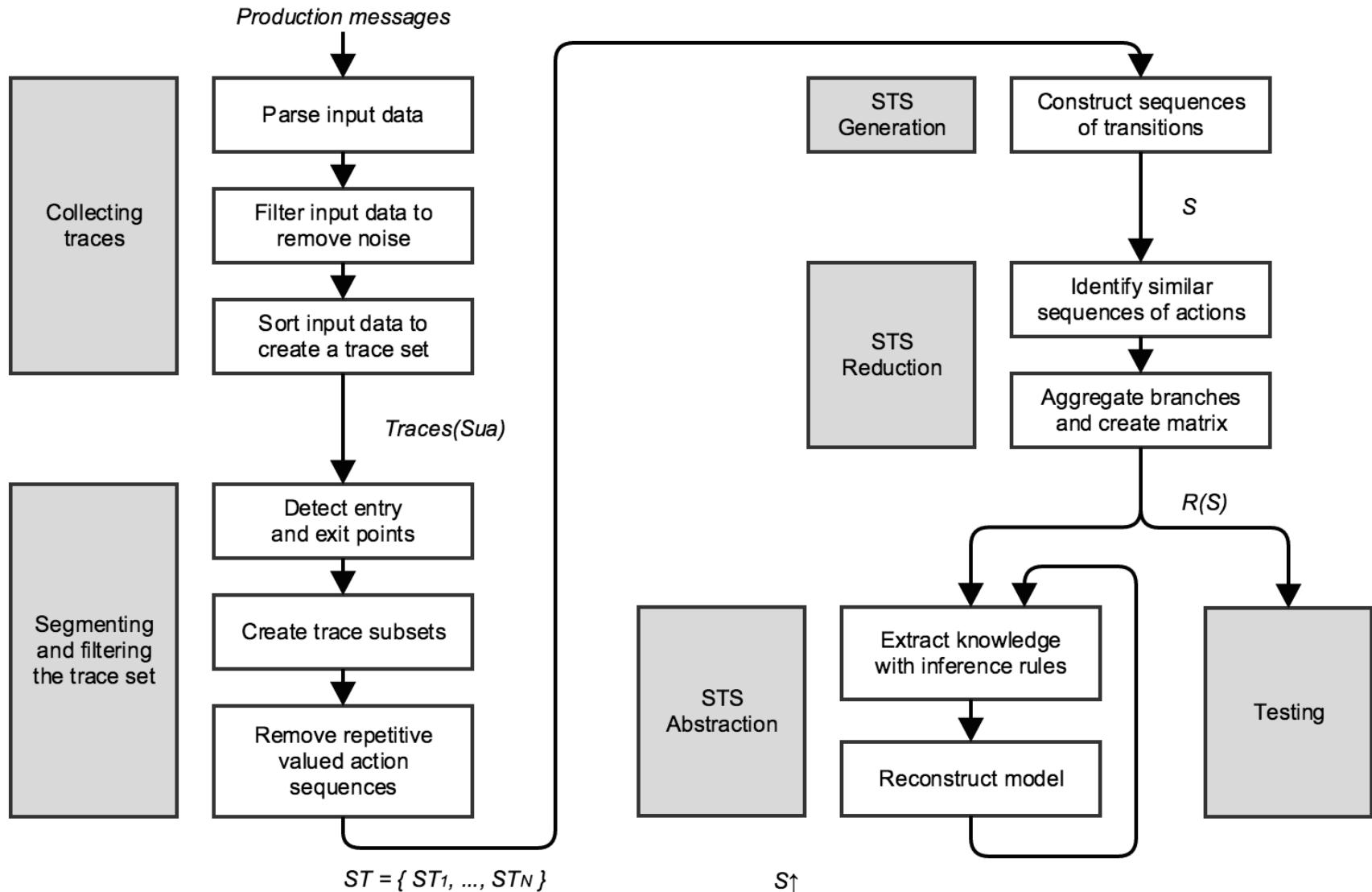
# Our Approach

By leveraging the information found in the production messages, we build **formal and exact models** (STS) that describe functional behaviours of a production system under analysis.

# The Big Picture



# Autofunk In Depth



# Autofunk

- Combines different fields: model inference, expert systems, and (now) machine learning
- Written in Java 8, reusing powerful libraries (e.g. [Spark](#), [Drools](#))
- More a **Proof of Concept** than a production-ready tool
- To be open sourced (no ETA yet)

# Experimentation

10 million production messages (20 days)



161,035 different traces



S	R(S)
77,058 branches	1,587 branches
43,536 branches	1,585 branches

❶ 2 entry points here

It took 5 minutes to build the two models.

# Work In Progress

# Offline Passive Testing

- Inferred models are used as specifications
- Another set of traces is collected on a system under test SUT (new or upgraded)



**Does SUT conforms to the specifications?**

# Conclusion

- **Fast and efficient** technique to infer formal models
- The more production messages, the better!
- But a few technical issues to tackle (memory consumption for instance)

# Future Work

- Deploying Autofunk as a real solution (WIP)
- Offline passive testing (WIP)
- Online passive testing

# Thank You. Questions?

